

Open Source Used In [talos] CUA: clamav-engine-go main-git-a0f81ec-38

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: 78EE117C99-2761621673

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this form.

In your requests please include the following reference number 78EE117C99-2761621673

Contents

1.1 lz4 1.9.3
1.1.1 Available under license
1.2 zlib 1.3.1
1.2.1 Available under license
1.3 zstd 1.3.1
1.3.1 Available under license
1.4 librdkafka 2.10.0-r0
1.4.1 Notifications
1.4.2 Available under license
1.5 openssl 3.0.15
1.5.1 Available under license
1.6 golang 1.25.0
1.6.1 Available under license
1.7 curl 8.14.1-r2
1.7.1 Available under license
1.8 skalibs 2.14.4.0-r0
1.8.1 Available under license
1.9 github.com/aws/aws-sdk-go-v2/service/ssooidc 1.35.3
1.9.1 Available under license
1.10 github.com/aws/aws-sdk-go-v2/service/sso 1.29.8
1.10.1 Available under license
1.11 goautoneg 0.0.0-20191010083416-a7dc8b61c822
1.11.1 Available under license
1.12 smithy-go 1.23.1
1.12.1 Available under license
1.13 aws-sdk-go-v2-internal-v4a 1.4.11

1.13.1 Available under license
1.14 libmd 1.1.0-r0
1.14.1 Available under license
1.15 github.com/baruwa-enterprise/clamd 1.0.1
1.15.1 Available under license
1.16 pax-utils 1.3.8-r1
1.16.1 Available under license
1.17 github.com/aws/aws-sdk-go-v2/credentials 1.18.19
1.17.1 Available under license
1.18 libcap 2.76-r0
1.18.1 Available under license
1.19 github.com/aws/aws-sdk-go-v2/service/internal/s3shared 1.19.11
1.19.1 Available under license
1.20 perks 1.0.1
1.20.1 Available under license
1.21 aws-sdk-go-v2-service-internal-accept-encoding 1.13.2
1.21.1 Available under license
1.22 github.com/confluentinc/confluent-kafka-go/v2 2.12.0
1.22.1 Available under license
1.23 github.com/aws/aws-sdk-go-v2/service/internal/checksum 1.9.2
1.23.1 Available under license
1.24 zlib 1.3.1-r2
1.24.1 Available under license
1.25 scanelf 1.3.8-r1
1.25.1 Available under license
1.26 github.com/aws/aws-sdk-go-v2/internal/configsources 1.4.11
1.26.1 Available under license
1.27 github.com/aws/aws-sdk-go-v2/feature/ec2/imds 1.18.11
1.27.1 Available under license
1.28 alpine-base 3.22.2-r0
1.28.1 Available under license
1.29 alpine-keys 2.5-r0
1.29.1 Available under license
1.30 logrus 1.9.3
1.30.1 Available under license
1.31 aws-sdk-go-v2-service-internal-presigned-url 1.13.11
1.31.1 Available under license
1.32 aws-sdk-go-v2-feature-ec2-imds 1.18.11
1.32.1 Available under license

1.33 protobuf 1.36.10
1.33.1 Available under license
1.34 aws-sdk-go-v2-internal-endpoints 2.7.11
1.34.1 Available under license
1.35 utmps 0.1.3.1-r0
1.35.1 Available under license
1.36 ca-certificates 20250911-r0
1.36.1 Available under license
1.37 aws-sdk-go-v2 1.39.4
1.37.1 Available under license
1.38 aws-sdk-go-v2-service-sts 1.38.9
1.38.1 Available under license
1.39 aws-sdk-go-v2-config 1.31.15
1.39.1 Available under license
1.40 libssl3 3.5.4-r0
1.40.1 Available under license
1.41 musl 1.2.5-r10
1.41.1 Available under license
1.42 aws-sdk-go-v2-internal-ini 1.8.4
1.42.1 Available under license
1.43 protobuf-go 1.36.10
1.43.1 Available under license
1.44 github.com/aws/aws-sdk-go-v2/aws/protocol/eventstream 1.7.2
1.44.1 Available under license
1.45 github.com/aws/aws-sdk-go-v2/service/internal/accept-encoding 1.13.2
1.45.1 Available under license
1.46 iputils-clockdiff 20240905-r0
1.46.1 Available under license
1.47 github.com/aws/aws-sdk-go-v2/config 1.31.15
1.47.1 Available under license
1.48 iputils-arping 20240905-r0
1.48.1 Available under license
1.49 apk-tools 2.14.9-r3
1.49.1 Available under license
1.50 kelseyhightower-envconfig 1.4.0
1.50.1 Available under license
1.51 go.yaml.in/yaml/v2 2.4.3
1.51.1 Available under license
1.52 aws-sdk-go-v2-service-sso 1.29.8

1.52.1 Available under license
1.53 openssl 3.5.4-r0
1.53.1 Available under license
1.54 aws-sdk-go-v2-aws-protocol-eventstream 1.7.2
1.54.1 Available under license
1.55 bash 5.2.37-r0
1.55.1 Available under license
1.56 go.yaml.in-yaml 2.4.3
1.56.1 Available under license
1.57 aws-sdk-go-v2-credentials 1.18.19
1.57.1 Available under license
1.58 pam 1.7.0-r4
1.58.1 Available under license
1.59 readline 8.2.13-r1
1.59.1 Available under license
1.60 ca-certificates-bundle 20250911-r0
1.60.1 Available under license
1.61 busybox 1.37.0-r19
1.61.1 Available under license
1.62 shadow 4.17.3-r0
1.62.1 Available under license
1.63 libncursesw 6.5_p20250503-r0
1.63.1 Available under license
1.64 ncurses 6.5_p20250503-r0
1.64.1 Available under license
1.65 utmps-libs 0.1.3.1-r0
1.65.1 Available under license
1.66 iputils-ping 20240905-r0
1.66.1 Available under license
1.67 github.com/aws/aws-sdk-go-v2/service/internal/presigned-url 1.13.11
1.67.1 Available under license
1.68 aws-sdk-go-v2-internal-configsources 1.4.11
1.68.1 Available under license
1.69 libcrypto3 3.5.4-r0
1.69.1 Available under license
1.70 alpine-baselayout-data 3.7.0-r0
1.70.1 Available under license
1.71 github.com/aws/aws-sdk-go-v2/internal/v4a 1.4.11
1.71.1 Available under license

1.72 aws-sdk-go-v2-service-internal-s3shared 1.19.11
1.72.1 Available under license
1.73 iputils-tracepath 20240905-r0
1.73.1 Available under license
1.74 iputils 20240905-r0
1.74.1 Available under license
1.75 alpine-baselayout 3.7.0-r0
1.75.1 Available under license
1.76 alpine-release 3.22.2-r0
1.76.1 Available under license
1.77 baruwa-enterprise-clamd 1.0.1
1.77.1 Available under license
1.78 libbsd 0.12.2-r0
1.78.1 Available under license
1.79 stdlib go1.25.0
1.79.1 Available under license
1.80 confluentinc-confluent-kafka-go 2.12.0
1.80.1 Available under license
1.81 github.com/aws/aws-sdk-go-v2/service/sts 1.38.9
1.81.1 Available under license
1.82 aws-sdk-go-v2-service-internal-checksum 1.9.2
1.82.1 Available under license
1.83 ssl-client 1.37.0-r19
1.83.1 Available under license
1.84 libapk2 2.14.9-r3
1.84.1 Available under license
1.85 github.com/aws/aws-sdk-go-v2/internal/endpoints/v2 2.7.11
1.85.1 Available under license
1.86 clientmodel 0.6.2
1.86.1 Available under license
1.87 skalibs-libs 2.14.4.0-r0
1.87.1 Available under license
1.88 aws-sdk-go-v2-service-ssooidc 1.35.3
1.88.1 Available under license
1.89 github.com/aws/smithy-go 1.23.1
1.89.1 Available under license
1.90 musl-utils 1.2.5-r10
1.90.1 Available under license
1.91 x-sys 0.37.0

1.91.1 Available under license
1.92 github.com/aws/aws-sdk-go-v2/internal/ini 1.8.4
1.92.1 Available under license
1.93 sirupsen-logrus 1.9.3
1.93.1 Available under license
1.94 aws-sdk-go-v2-service-s3 1.89.0
1.94.1 Available under license
1.95 common 0.67.2
1.95.1 Available under license
1.96 prometheus-common 0.67.2
1.96.1 Available under license
1.97 cjson 1.7.19-r0
1.97.1 Notifications
1.97.2 Available under license
1.98 prometheus-client-model 0.6.2
1.98.1 Available under license
1.99 prometheus-client 1.23.2
1.99.1 Available under license
1.100 xxhash 2.3.0
1.100.1 Available under license
1.101 procfs 0.19.2
1.101.1 Available under license
1.102 aws-sdk-go-v2-feature-s3-manager 1.20.1
1.102.1 Available under license
1.103 github.com/aws/aws-sdk-go-v2/service/s3 1.89.0
1.103.1 Available under license
1.104 github.com/aws/aws-sdk-go-v2/feature/s3/manager 1.20.1
1.104.1 Available under license

1.1 lz4 1.9.3

1.1.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which
gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and

modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work
are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer

to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution
and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source
code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing
or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to

this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing
version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it

free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU
General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.
Copyright (c) 2014, lpsantil
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
This repository uses 2 different licenses :
- all files in the `lib` directory use a BSD 2-Clause license
- all other files use a GPLv2 license, unless explicitly stated otherwise

Relevant license is reminded at the top of each source file,
and with presence of COPYING or LICENSE file in associated directories.

This model is selected to emphasize that
files in the `lib` directory are designed to be included into 3rd party applications,
while all other files, in `programs`, `tests` or `examples`,
receive more limited attention and support for such scenario.
LZ4 Library
Copyright (c) 2011-2016, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
 list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
 list of conditions and the following disclaimer in the documentation and/or
 other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
 GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: liblz4
Upstream-Contact: Yann Collet <Cyan4973@github.com>
Source: https://github.com/lz4/lz4

Files: *
Copyright: (C) 2011-2020 Yann Collet
License: GPL-2+
The full text of license: https://github.com/lz4/lz4/blob/dev/lib/LICENSE

1.2 zlib 1.3.1

1.2.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

 This software is provided 'as-is', without any express or implied
 warranty. In no event will the authors be held liable for any damages
 arising from the use of this software.

 Permission is granted to anyone to use this software for any purpose,
 including commercial applications, and to alter it and redistribute it
 freely, subject to the following restrictions:

 1. The origin of this software must not be misrepresented; you must not

claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu
/* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.3.1, January 22nd, 2024

Copyright (C) 1995-2024 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for
Comments) 1950 to 1952 in the files http://tools.ietf.org/html/rfc1950
(zlib format), rfc1951 (deflate format) and rfc1952 (gzip format).
*/
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including

the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
/* gzlib.c -- zlib functions common to reading and writing gzip files
* Copyright (C) 2004-2024 Mark Adler
* For conditions of distribution and use, see copyright notice in zlib.h
*/

1.3 zstd 1.3.1

1.3.1 Available under license :

BSD-style without advertising clause

1.4 librdkafka 2.10.0-r0

1.4.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).
This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.4.2 Available under license :

For the files wingetopt.c wingetopt.h downloaded from https://github.com/alex85k/wingetopt

/*
* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN

* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory,
Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*-
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/
src/rdxxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library
Copyright (c) 2011-2020, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
 list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
 list of conditions and the following disclaimer in the documentation and/or
 other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:
  chongo <Landon Curt Noll> /\oo/\
  http://www.isthe.com/chongo/

Share and Enjoy! :-)
regexp.c and regexp.h from https://github.com/ccxvii/minilibs sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"
These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.
"
FindZstd.cmake: git@github.com:facebook/folly.git 87f1a403b49552dae75ae94c8610dd5979913477

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership
of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including
but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object
form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a
readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications
and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for
reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed
to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if
You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
For src/crc32c.c copied (with modifications) from
http://stackoverflow.com/a/17646775/1821055

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
This software is provided 'as-is', without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original
software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler
madler@alumni.caltech.edu
*/
For sys/queue.h:

* Copyright (c) 1991, 1993
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 4. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* $FreeBSD$

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
LICENSE
---
librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill
2023, Confluent Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished
to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

For src/crc32c.c copied (with modifications) from
http://stackoverflow.com/a/17646775/1821055

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
This software is provided 'as-is',
without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler
madler@alumni.caltech.edu
*/

LICENSE.fnv1a

---

parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please
do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:
  chongo <Landon Curt Noll> /\oo/\
  http://www.isthe.com/chongo/

Share and Enjoy! :-)

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram https://github.com/codahale/hdrhistogram
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License
(MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE

LICENSE.lz4
---
src/rdxxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library
Copyright (c) 2011-2020, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.murmur2

---

parts of src/rdmurmur2.c: git@github.com:abrandoned/murmur2.git

MurMurHash2 Library

//---

// MurmurHash2 was written by Austin Appleby, and is placed in the public

// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person
obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.nanopb

For
files in src/nanopb :
https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or
implied warranty. In no event will the authors be held liable

for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you
must not claim that you wrote the original software. If you use
this software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may
not be removed or altered from any source
distribution.

LICENSE.opentelemetry

For files in src/opentelemetry: https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes
of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution
incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE
text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding
such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event
and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent
with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.

==========

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted,
 free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

LICENSE.queue

For
 sys/queue.h:

* Copyright (c) 1991, 1993
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 4. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR
 IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* $FreeBSD$

LICENSE.regexp
---
regexp.c and regexp.h from https://github.com/ccxvii/minilibs sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"
These libraries are in the public domain (or the equivalent where that is not
 possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.
"

LICENSE.snappy

##
LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h
originally retrieved from http://github.com/andikleen/snappy-c
git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219
##

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions
 in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.tinycthread

From https://github.com/tinycthread/tinycthread/README.txt c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from https://github.com/alex85k/wingetopt

/*
* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*-
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice,
this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/
##
LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h
originally retrieved from http://github.com/andikleen/snappy-c
git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219
##

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Intel
 Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill
          2023, Confluent Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: librdkafka
Source: https://github.com/confluentinc/librdkafka

License: BSD-2-clause
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
.
1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.
.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: *
Copyright: 2012-2022, Magnus Edenhill; 2023, Confluent Inc.

License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h
Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>
License: MIT
 Permission is hereby granted, free of charge, to any person obtaining a copy
 of this software and associated documentation files (the "Software"), to deal
 in the Software without restriction, including without limitation the rights
 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 copies of the Software, and to
 permit persons to whom the Software is
 furnished to do so, subject to the following conditions:
 .
 The above copyright notice and this permission notice shall be included in
 all copies or substantial portions of the Software.
 .
 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
 THE SOFTWARE.

Files: src/snappy.c src/snappy.h src/snappy_compat.h
Copyright: 2005, Google Inc.
 2011, Intel Corporation
License: BSD-3-clause
 Redistribution and use in source and binary forms, with or without
 modification, are permitted provided that the following conditions are
 met:
 .
 * Redistributions of source code must retain the above copyright
 notice, this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above
 copyright notice, this list of conditions and the following disclaimer
 in the documentation and/or other materials provided with the
 distribution.
 * Neither the name of Google Inc. nor the names of its
 contributors may be used to endorse or promote products derived from
 this software without specific prior written permission.
 .
 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: src/rdxxhash.h src/rdxxhash.c
Copyright: 2012-2014, Yann Collet
License: BSD-2-clause

Files: src/regexp.h src/regexp.c
Copyright: n/a
License: public-domain

License: public-domain
The files tagged with this license contain the following paragraphs:
.
These libraries are in the public domain (or the equivalent where that is not
possible). You can do anything you want with them. You have no legal
obligation to do anything else, although I appreciate attribution.

Files: debian/*
Copyright: 2013 Faidon Liambotis <paravoid@debian.org>
License: BSD-2-clause
BSD-style without advertising clause
Copyright (c) 2016-2019, Magnus Edenhill
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
For files in src/nanopb :
https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or
implied warranty. In no event will the authors be held liable
for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you
must not claim that you wrote the original software. If you use
this software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source

distribution.
parts of src/rdmurmur2.c: git@github.com:abrandoned/murmur2.git

MurMurHash2 Library
//---
// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: librdkafka
Source: https://github.com/confluentinc/librdkafka

License: BSD-2-clause
 Redistribution and use in source and binary forms, with or without
 modification, are permitted provided that the following conditions are met:
 .
 1. Redistributions of source code must retain the above copyright notice,
    this list of conditions and the following disclaimer.
 2. Redistributions in binary form must reproduce the above copyright notice,
    this list of conditions and the following disclaimer in the documentation
    and/or other materials provided with the distribution.
 .
 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 LIABLE FOR
 ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

Files: *
Copyright: 2012-2022, Magnus Edenhill; 2023 Confluent Inc.
License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h
Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>
License: MIT
 Permission is hereby granted, free of charge, to any person obtaining a copy
 of this software and associated documentation files (the "Software"), to deal
 in the Software without restriction, including without limitation the rights
 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 copies of the Software, and to permit
 persons to whom the Software is
 furnished to do so, subject to the following conditions:
 .
 The above copyright notice and this permission notice shall be included in
 all copies or substantial portions of the Software.

.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files: src/snappy.c src/snappy.h src/snappy_compat.h
Copyright: 2005, Google Inc.
2011, Intel Corporation
License: BSD-3-clause
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
.
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.
.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: debian/*
Copyright: 2013 Faidon Liambotis <paravoid@debian.org>
License: BSD-2-clause
This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram https://github.com/codahale/hdrhistogram
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License (MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE
From https://github.com/tinycthread/tinycthread/README.txt c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
        2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.

3. This notice may not be removed or altered from

any source
distribution.
The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.
==========

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.5 openssl 3.0.15

1.5.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under
the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

a) the GNU General Public License as published by the Free
Software Foundation; either version 1, or (at your option) any
later version, or
b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so

that any problems introduced by others will not reflect on the original
authors' reputations.

The precise terms and conditions
for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which
contains a notice placed by the copyright holder saying it may be
distributed under the terms of this General Public License. The
"Program", below, refers to any such program or work, and a "work based
on the Program" means either the Program or any work containing the
Program or a portion of it, either verbatim or with modifications. Each
licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source
code as you receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice and
disclaimer of warranty; keep intact all the notices that refer to this
General Public License and to the absence of any warranty; and give any
other recipients of the
Program a copy of this General Public License
along with the Program. You may charge a fee for the physical act of
transferring a copy.

2. You may modify your copy or copies of the Program or any portion of
it, and copy and distribute such modifications under the terms of Paragraph
1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that
you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that
in whole or in part contains the Program or any part thereof, either
with or without modifications, to be licensed at no charge to all
third parties under the terms of this General Public License (except
that you may choose to grant warranty protection to some or all
third parties, at your option).

c) If the modified program normally reads commands interactively when
run, you must cause it, when started running for such interactive
use
in the simplest and most usual way, to print or display an
announcement including an appropriate copyright notice and a notice
that there is no warranty (or else, saying that you provide a

warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties

remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free
Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter
the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright

Holder, and derivatives of that collection
of files created through
textual modification.
- "Standard Version" refers to such a Package if it has not been modified,
or has been modified in accordance with the wishes of the Copyright
Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for
the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media
cost, duplication charges, time of people involved, and so on. (You will
not be required to justify it to the Copyright Holder, but only to the
computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though
there may be fees involved in handling the item. It also means that
recipients of the item may redistribute it under the same conditions they
received it.

1. You may make and give away verbatim copies of
the source form of the
Standard Version of this Package without restriction, provided that you
duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived
from the Public Domain or from the Copyright Holder. A Package modified in such
a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that
you insert a prominent notice in each changed file stating how and when you
changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them
Freely Available, such as by posting said modifications to Usenet or an
equivalent medium, or placing the modifications on a major archive site
such as ftp.uu.net, or by allowing the Copyright Holder to include your
modifications in the Standard Version of the Package.

b) use the modified
Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with
standard executables, which must also be provided, and provide a separate
manual page for each non-standard executable that clearly documents how it
differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable

form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files,
together with instructions (in the manual page or equivalent) on where to
get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package
with your modifications.

c) accompany any non-standard executables with their corresponding Standard
Version executables, giving the non-standard executables non-standard
names,
and clearly documenting the differences in manual pages (or
equivalent), together with instructions on where to get the Standard
Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this
Package. You may charge any fee you choose for support of this Package. You
may not charge a fee for this Package itself. However, you may distribute this
Package in aggregate with other (possibly commercial) programs as part of a
larger (possibly commercial) software distribution provided that you do not
advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output
from the programs of this Package do not automatically fall under the copyright
of this Package, but belong to whomever generated them, and may be sold
commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package
shall not
be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote
products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.6 golang 1.25.0

1.6.1 Available under license :

```
// Copyright 2024 The Go Authors. All rights reserved.
// Use of this source code is governed by a BSD-style
// license that can be found in the LICENSE file.

package copyright

import (
 "bytes"
```

```
 "internal/testenv"
 "io"
 "io/fs"
 "os"
 "path/filepath"
 "testing"
)

var copyright = []byte("Copyright")

var permitted = [][]byte{
 []byte("// Code generated by "),
 []byte("// Code generated from "),
 []byte("// Created by cgo -cdefs"),
 []byte("// DO NOT EDIT\n// generated by:"),
 []byte("// Empty assembly file"),
 []byte("// Generated using cgo"),
 []byte("// Original source:\n//\thttp://www.zorinaq.com/papers/md5-amd64.html"), // public domain crypto/md5
 []byte("// created by cgo -cdefs"),
 []byte("// go run mkasm.go"),
 []byte("// mkerrors"),
 []byte("// mksys"),
 []byte("// run\n// Code generated by"), // cmd/compile/internal/test/constFold_test.go
}

func TestCopyright(t *testing.T) {
 buf := make([]byte, 2048)
 filepath.WalkDir(filepath.Join(testenv.GOROOT(t), "src"), func(path string,
 d fs.DirEntry, err error) error {
  if d.IsDir() && (d.Name() == "testdata" || d.Name() == "vendor") {
   return filepath.SkipDir
  }
  switch filepath.Ext(d.Name()) {
  default:
   return nil
  case ".s", ".go":
   // check
  }

  f, err := os.Open(path)
  if err != nil {
   t.Error(err)
   return nil
  }
  defer f.Close()
  n, err := f.Read(buf)
  if err != nil && err != io.EOF {
   t.Error(err)
```

```
  return nil
 }
 b := buf[:n]
 if bytes.Contains(b, copyright) {
  return nil
 }
 for _, ok := range permitted {
  if bytes.HasPrefix(b, ok) {
   return nil
  }
 }
 t.Errorf("%s: missing copyright notice", path)
 return nil
})
}
```

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google LLC nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE
 FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The Go source code and supporting files in this directory
are covered by the usual Go license (see ../../../../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto_linux_amd64.syso object file is built
from BoringSSL source code by build/build.sh and is covered
by the BoringSSL license reproduced below and also at
https://boringssl.googlesource.com/boringssl/+/fips-20220613/LICENSE.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL
licensing. Files that are completely new have a Google copyright and an ISC
license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:
https://cla.developers.google.com/clas

Some files from Intel are under yet another license, which is also included
underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the
OpenSSL License and the original SSLeay license apply to the toolkit.
 See below
for the actual license texts. Actually both licenses are BSD-style Open Source
licenses. In case of any license issues related to OpenSSL please contact
openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from
some authors is recorded for use of their work. (This is purely for our own
record keeping.)
 27287199
 27287880
 27287883

 OpenSSL License

/* ==
* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following
 disclaimer in
*   the documentation and/or other materials provided with the
*   distribution.
*

* 3. All advertising materials mentioning features or use of this
*   software must display the following acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*   endorse or promote products derived from this software without
*   prior written permission. For written permission, please contact
*   openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*   nor may "OpenSSL" appear in their names without prior written
*   permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit
 (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product
 includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written

* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of
 the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are
 not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE

POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.
*
* Permission to use, copy, modify, and/or distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
* OF CONTRACT, NEGLIGENCE
 OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

Some files from Intel carry the following license:

Copyright (c) 2012, Intel Corporation
#
All rights reserved.
#
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
#
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
#
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.
#
* Neither the name of the Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from

this software without specific prior written permission.
#
#
THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION
 ""AS IS"" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.
#
https://developers.google.com/open-source/cla/individual
https://developers.google.com/open-source/cla/corporate
#
Names should be added to this file as:
Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyoun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its

contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the
authors and should not be interpreted as representing official policies, either expressed
or implied, of Andrea Leofreddi.

1.7 curl 8.14.1-r2

1.7.1 Available under license :

Copyright (c) <year> <owner>.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
COPYRIGHT AND PERMISSION NOTICE

Copyright (C) Daniel Stenberg, <daniel@haxx.se>, and many
contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other dealings
in this
Software without prior written authorization of the copyright holder.
BSD-4-Clause (University of California-Specific)

Copyright [various years] The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific
prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS

OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM
DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL
INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING
FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2025, Daniel Stenberg, <daniel@haxx.se>, and many
contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other
dealings
in this Software without prior written authorization of the copyright holder.

1.8 skalibs 2.14.4.0-r0

1.8.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed
to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work
which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim
copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably
considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work

based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code
or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy
the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.
Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free
redistribution of the Program by
all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through
any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version
number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE
LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands
`show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.
Automatically generated by apkbuild-cpan, template 4
Contributor: Valery Kartel <valery.kartel@gmail.com>
Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
  INSTALLDIRS=vendor \

```
 NO_PACKLIST=1 \
 NO_PERLLOCAL=1
 make
}

check() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 make test
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
  BSD-Resource-1.2911.tar.gz
"
# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
 0001-bsd-games-scores-dir.patch
 "
options="!check" # no tests

build() {
 # some GNU autoconf options are ignored, but it works
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr \
  --sysconfdir=/etc \
  --mandir=/usr/share/man \
  --scoredir=/usr/share/$pkgname
 make
}
```

```
package() {
 make DESTDIR="$pkgdir" install
 install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
  bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6  0001-bsd-games-scores-dir.patch
"
```

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION
 WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check"  # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2
 "

builddir="$srcdir"/libart_lgpl-$pkgver
```

```
prepare() {
 update_config_sub
 default_prepare
}

build() {
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr
 make
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020  libart_lgpl-2.3.21.tar.bz2"
```

mini_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>
Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

```
pkgname=spdx-licenses
pkgver=3.26.0
pkgrel=0
pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdfnt rdfturtle rdfxml template text"

for type in $_types; do
 subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
 mkdir -p "$pkgdir"
}

_subpkg() {
 local type=${subpkgname/$pkgname-/}
 pkgdesc="$pkgdesc ($type)"
 install_if="$pkgname=$pkgver-r$pkgrel"
 mkdir -p "$subpkgdir"/usr/share/spdx
 cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}

list() {
 pkgdesc="$pkgdesc (licence list)"
 mkdir -p "$subpkgdir"/usr/share/spdx
 local i; for i in "$builddir"/text/*.txt;
 do
  local license=${i##*/}
  echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
 done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a  license-list-data-3.26.0.tar.gz
"
# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
```

```
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
 cdefs.h
 queue.h
 tree.h
"
builddir="$srcdir"
options="!check" # just headers

package() {
 mkdir -p "$pkgdir"
 install -Dm644 -t "$pkgdir"/usr/include/sys \
  cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178  cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750  queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3  tree.h
"
```

Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)
The text rotation code in w_rottext.c was written by Alan Richardson.
 The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker
The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick
The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:
"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95
http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html
Contributor: Dhruvin Gandhi <contact@dhru.vin>
Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=py3-license-expression
_pkgname=license_expression
pkgver=30.4.1
pkgrel=0
pkgdesc="Library to parse, compare, simplify and normalize license expressions"
url="https://github.com/nexB/license-expression"
arch="noarch"
license="Apache-2.0"
depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir/$_pkgname-$pkgver"

build() {
 gpep517 build-wheel \
  --wheel-dir .dist \
  --output-fd 3 3>&1 >&2
}

check() {
 python3 -m venv --clear --without-pip --system-site-packages .testenv
 .testenv/bin/python3 -m installer .dist/*.whl
 .testenv/bin/python3 -m pytest
}

package() {
 python3 -m installer -d "$pkgdir" \
  .dist/*.whl
}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312fda3f699cbc7c4fbd47a71e220440fb0
  license_expression-30.4.1.tar.gz
"
/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

```
/*
* Copyright (c) 1991, 1993
* The Regents of the University of California.  All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
*/

#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_

/*
* This file defines five types of data structures: singly-linked lists,
* lists, simple queues, tail queues, and circular queues.
*
* A singly-linked list is headed by a single forward pointer. The
* elements are singly linked for minimum space and
 pointer manipulation
* overhead at the expense of O(n) removal for arbitrary elements. New
* elements can be added to the list after an existing element or at the
* head of the list.  Elements being removed from the head of the list
* should use the explicit macro for this purpose for optimum
* efficiency. A singly-linked list may only be traversed in the forward
* direction.  Singly-linked lists are ideal for applications with large
```

```
* datasets and few or no removals or for implementing a LIFO queue.
*
* A list is headed by a single forward pointer (or an array of forward
* pointers for a hash table header). The elements are doubly linked
* so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before
* or after an existing element or at the head of the list. A list
* may only be traversed in the forward direction.
*
* A simple queue is headed by a pair of pointers, one the head of the
* list and the
 other to the tail of the list. The elements are singly
* linked to save space, so elements can only be removed from the
* head of the list. New elements can be added to the list after
* an existing element, at the head of the list, or at the end of the
* list. A simple queue may only be traversed in the forward direction.
*
* A tail queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before or
* after an existing element, at the head of the list, or at the end of
* the list. A tail queue may be traversed in either direction.
*
* A circle queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements
 can be added to the list before or after
* an existing element, at the head of the list, or at the end of the list.
* A circle queue may be traversed in either direction, but has a more
* complex end of list detection.
*
* For details on the use of these macros, see the queue(3) manual page.
*/

/*
* Include the definition of NULL only on NetBSD because sys/null.h
* is not available elsewhere.  This conditional makes the header
* portable and it can simply be dropped verbatim into any system.
* The caveat is that on other systems some other header
* must provide NULL before the macros can be used.
*/
#ifdef __NetBSD__
#include <sys/null.h>
#endif

#if defined(QUEUEDEBUG)
```

```
# if defined(_KERNEL)
#  define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
#  include <err.h>
#  define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
* Singly-linked List definitions.
*/
#define SLIST_HEAD(name, type)      \
struct name {       \
 struct type *slh_first; /* first
 element */   \
}

#define SLIST_HEAD_INITIALIZER(head)     \
 { NULL }

#define SLIST_ENTRY(type)      \
struct {       \
 struct type *sle_next; /* next element */   \
}

/*
* Singly-linked List access methods.
*/
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head)  NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field)     \
 for((var) = (head)->slh_first;     \
    (var) != SLIST_END(head);     \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = SLIST_FIRST((head));    \
    (var) != SLIST_END(head) &&     \
    ((tvar) = SLIST_NEXT((var), field), 1);   \
    (var) = (tvar))

/*
* Singly-linked List functions.
*/
#define SLIST_INIT(head) do {      \
```

```
 (head)->slh_first = SLIST_END(head);    \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do {   \
 (elm)->field.sle_next = (slistelm)->field.sle_next;  \
 (slistelm)->field.sle_next
 = (elm);    \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do {    \
 (elm)->field.sle_next = (head)->slh_first;   \
 (head)->slh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_AFTER(slistelm, field) do {    \
 (slistelm)->field.sle_next =     \
     SLIST_NEXT(SLIST_NEXT((slistelm), field), field);  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do {     \
 (head)->slh_first = (head)->slh_first->field.sle_next;  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do {   \
 if ((head)->slh_first == (elm)) {    \
  SLIST_REMOVE_HEAD((head), field);   \
 }       \
 else {       \
  struct type *curelm = (head)->slh_first;  \
  while(curelm->field.sle_next != (elm))   \
   curelm = curelm->field.sle_next;  \
  curelm->field.sle_next =    \
      curelm->field.sle_next->field.sle_next;  \
 }       \
} while (/*CONSTCOND*/0)


/*
* List definitions.
 */
#define LIST_HEAD(name, type)      \
struct name {       \
 struct type *lh_first; /* first element */   \
}

#define LIST_HEAD_INITIALIZER(head)     \
 { NULL }
```

```
#define LIST_ENTRY(type)      \
struct {        \
 struct type *le_next; /* next element */   \
 struct type **le_prev; /* address of previous next element */ \
}

/*
* List access methods.
*/
#define LIST_FIRST(head)  ((head)->lh_first)
#define LIST_END(head)   NULL
#define LIST_EMPTY(head)  ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field)  ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field)     \
 for ((var) = ((head)->lh_first);    \
     (var) != LIST_END(head);     \
     (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = LIST_FIRST((head));    \
     (var) != LIST_END(head) &&     \
     ((tvar) = LIST_NEXT((var), field), 1);   \
     (var) = (tvar))

#define LIST_MOVE(head1, head2) do {     \
 LIST_INIT((head2));      \
 if (!LIST_EMPTY((head1)))
 {     \
  (head2)->lh_first = (head1)->lh_first;   \
  LIST_INIT((head1));     \
 }        \
} while (/*CONSTCOND*/0)

/*
* List functions.
*/
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)   \
 if ((head)->lh_first &&      \
     (head)->lh_first->field.le_prev != &(head)->lh_first) \
  QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field)     \
 if ((elm)->field.le_next &&     \
     (elm)->field.le_next->field.le_prev !=   \
     &(elm)->field.le_next)     \
  QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm),  \
```

```
    __FILE__, __LINE__);    \
 if (*(elm)->field.le_prev != (elm))    \
  QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm),  \
    __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)    \
 (elm)->field.le_next = (void *)1L;    \
 (elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,
 field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do {      \
 (head)->lh_first = LIST_END(head);    \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do {   \
 QUEUEDEBUG_LIST_OP((listelm), field)    \
 if (((elm)->field.le_next = (listelm)->field.le_next) !=  \
    LIST_END(head))      \
  (listelm)->field.le_next->field.le_prev =  \
     &(elm)->field.le_next;    \
 (listelm)->field.le_next = (elm);    \
 (elm)->field.le_prev = &(listelm)->field.le_next;  \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do {   \
 QUEUEDEBUG_LIST_OP((listelm), field)    \
 (elm)->field.le_prev = (listelm)->field.le_prev;  \
 (elm)->field.le_next = (listelm);    \
 *(listelm)->field.le_prev = (elm);    \
 (listelm)->field.le_prev = &(elm)->field.le_next;  \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do {    \
 QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field)  \
 if (((elm)->field.le_next
 = (head)->lh_first) != LIST_END(head))\
  (head)->lh_first->field.le_prev = &(elm)->field.le_next;\
 (head)->lh_first = (elm);     \
 (elm)->field.le_prev = &(head)->lh_first;   \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do {     \
 QUEUEDEBUG_LIST_OP((elm), field)    \
 if ((elm)->field.le_next != NULL)    \
```

```
 (elm)->field.le_next->field.le_prev =    \
     (elm)->field.le_prev;    \
 *(elm)->field.le_prev = (elm)->field.le_next;   \
 QUEUEDEBUG_LIST_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do {    \
 if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
  (elm2)->field.le_next->field.le_prev =   \
     &(elm2)->field.le_next;    \
 (elm2)->field.le_prev = (elm)->field.le_prev;   \
 *(elm2)->field.le_prev = (elm2);    \
 QUEUEDEBUG_LIST_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)

/*
* Simple queue definitions.
*/
#define SIMPLEQ_HEAD(name, type)     \
struct name {        \
 struct
 type *sqh_first; /* first element */   \
 struct type **sqh_last; /* addr of last next element */  \
}

#define SIMPLEQ_HEAD_INITIALIZER(head)     \
 { NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type)      \
struct {        \
 struct type *sqe_next; /* next element */   \
}

/*
* Simple queue access methods.
*/
#define SIMPLEQ_FIRST(head)  ((head)->sqh_first)
#define SIMPLEQ_END(head)  NULL
#define SIMPLEQ_EMPTY(head)  ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->sqh_first);    \
     (var) != SIMPLEQ_END(head);     \
     (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next)   \
 for ((var) = ((head)->sqh_first);    \
```

```
    (var) != SIMPLEQ_END(head) &&    \
    ((next = ((var)->field.sqe_next)), 1);   \
    (var) = (next))

/*
* Simple queue functions.
*/
#define SIMPLEQ_INIT(head) do {      \
 (head)->sqh_first = NULL;     \
 (head)->sqh_last
 = &(head)->sqh_first;    \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do {   \
 if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
  (head)->sqh_last = &(elm)->field.sqe_next;  \
 (head)->sqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do {   \
 (elm)->field.sqe_next = NULL;     \
 *(head)->sqh_last = (elm);     \
 (head)->sqh_last = &(elm)->field.sqe_next;   \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL)\
  (head)->sqh_last = &(elm)->field.sqe_next;  \
 (listelm)->field.sqe_next = (elm);    \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do {    \
 if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
  (head)->sqh_last = &(head)->sqh_first;   \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do {   \
 if
 (((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
    == NULL)       \
  (head)->sqh_last = &(elm)->field.sqe_next;  \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do {   \
 if ((head)->sqh_first == (elm)) {    \
  SIMPLEQ_REMOVE_HEAD((head), field);   \
 } else {       \
  struct type *curelm = (head)->sqh_first;  \
```

```
  while (curelm->field.sqe_next != (elm))   \
   curelm = curelm->field.sqe_next;  \
  if ((curelm->field.sqe_next =    \
   curelm->field.sqe_next->field.sqe_next) == NULL) \
      (head)->sqh_last = &(curelm)->field.sqe_next; \
 }       \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do {    \
 if (!SIMPLEQ_EMPTY((head2))) {     \
  *(head1)->sqh_last = (head2)->sqh_first;  \
  (head1)->sqh_last = (head2)->sqh_last;  \
  SIMPLEQ_INIT((head2));     \
 }       \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field)     \
 (SIMPLEQ_EMPTY((head)) ?      \
  NULL :       \
        ((struct type
 *)(void *)    \
  ((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
* Tail queue definitions.
*/
#define _TAILQ_HEAD(name, type, qual)     \
struct name {        \
 qual type *tqh_first;  /* first element */  \
 qual type *qual *tqh_last; /* addr of last next element */ \
}
#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head)     \
 { TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual)     \
struct {        \
 qual type *tqe_next;  /* next element */  \
 qual type *qual *tqe_prev; /* address of previous next element */\
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
* Tail queue access methods.
*/
#define TAILQ_FIRST(head)  ((head)->tqh_first)
#define TAILQ_END(head)   (NULL)
```

```
#define TAILQ_NEXT(elm, field)  ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
 (*(((struct headname *)(void *)((head)->tqh_last))->tqh_last))
#define TAILQ_PREV(elm, headname, field) \
 (*(((struct headname
 *)(void *)((elm)->field.tqe_prev))->tqh_last))
#define TAILQ_EMPTY(head)  (TAILQ_FIRST(head) == TAILQ_END(head))


#define TAILQ_FOREACH(var, head, field)     \
 for ((var) = ((head)->tqh_first);    \
     (var) != TAILQ_END(head);     \
     (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next)   \
 for ((var) = ((head)->tqh_first);    \
     (var) != TAILQ_END(head) &&     \
     ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field)  \
 for ((var) = TAILQ_LAST((head), headname);   \
     (var) != TAILQ_END(head);     \
     (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
 for ((var) = TAILQ_LAST((head), headname);   \
     (var) != TAILQ_END(head) &&     \
     ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
* Tail queue functions.
*/
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head,
 elm, field)   \
 if ((head)->tqh_first &&     \
     (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
  QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)   \
 if (*(head)->tqh_last != NULL)     \
  QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field)     \
 if ((elm)->field.tqe_next &&     \
     (elm)->field.tqe_next->field.tqe_prev !=   \
     &(elm)->field.tqe_next)     \
  QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
```

```
    __FILE__, __LINE__);    \
 if (*(elm)->field.tqe_prev != (elm))    \
  QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
    __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)   \
 if ((elm)->field.tqe_next == NULL &&    \
    (head)->tqh_last != &(elm)->field.tqe_next)   \
  QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d",\
     (head), (elm), __FILE__,
 __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)    \
 (elm)->field.tqe_next = (void *)1L;    \
 (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do {      \
 (head)->tqh_first = TAILQ_END(head);    \
 (head)->tqh_last = &(head)->tqh_first;    \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do {   \
 QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field)  \
 if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head))\
  (head)->tqh_first->field.tqe_prev =   \
     &(elm)->field.tqe_next;    \
 else       \
  (head)->tqh_last = &(elm)->field.tqe_next;  \
 (head)->tqh_first = (elm);     \
 (elm)->field.tqe_prev = &(head)->tqh_first;   \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head,
 elm, field) do {   \
 QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field)  \
 (elm)->field.tqe_next = TAILQ_END(head);   \
 (elm)->field.tqe_prev = (head)->tqh_last;   \
 *(head)->tqh_last = (elm);     \
 (head)->tqh_last = &(elm)->field.tqe_next;   \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 QUEUEDEBUG_TAILQ_OP((listelm), field)    \
```

```
 if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
     TAILQ_END(head))      \
  (elm)->field.tqe_next->field.tqe_prev =   \
      &(elm)->field.tqe_next;    \
 else       \
  (head)->tqh_last = &(elm)->field.tqe_next;  \
 (listelm)->field.tqe_next = (elm);    \
 (elm)->field.tqe_prev = &(listelm)->field.tqe_next;  \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do {   \
 QUEUEDEBUG_TAILQ_OP((listelm), field)    \
 (elm)->field.tqe_prev = (listelm)->field.tqe_prev;  \
 (elm)->field.tqe_next = (listelm);    \
 *(listelm)->field.tqe_prev = (elm);    \
 (listelm)->field.tqe_prev
 = &(elm)->field.tqe_next;  \
} while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do {    \
 QUEUEDEBUG_TAILQ_PREREMOVE((head), (elm), field)  \
 QUEUEDEBUG_TAILQ_OP((elm), field)    \
 if (((elm)->field.tqe_next) != TAILQ_END(head))   \
  (elm)->field.tqe_next->field.tqe_prev =   \
     (elm)->field.tqe_prev;    \
 else       \
  (head)->tqh_last = (elm)->field.tqe_prev;  \
 *(elm)->field.tqe_prev = (elm)->field.tqe_next;   \
 QUEUEDEBUG_TAILQ_POSTREMOVE((elm), field);   \
} while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do {   \
      if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
    TAILQ_END(head))        \
            (elm2)->field.tqe_next->field.tqe_prev =  \
               &(elm2)->field.tqe_next;    \
      else       \
            (head)->tqh_last = &(elm2)->field.tqe_next;  \
      (elm2)->field.tqe_prev = (elm)->field.tqe_prev;   \
      *(elm2)->field.tqe_prev = (elm2);    \
 QUEUEDEBUG_TAILQ_POSTREMOVE((elm),
 field);   \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do {    \
 if (!TAILQ_EMPTY(head2)) {     \
  *(head1)->tqh_last = (head2)->tqh_first;  \
  (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
```

```
 (head1)->tqh_last = (head2)->tqh_last;   \
 TAILQ_INIT((head2));     \
 }       \
} while (/*CONSTCOND*/0)

/*
* Singly-linked Tail queue declarations.
*/
#define STAILQ_HEAD(name, type)      \
struct name {        \
 struct type *stqh_first; /* first element */  \
 struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head)     \
 { NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type)      \
struct {        \
 struct type *stqe_next; /* next element */   \
}

/*
* Singly-linked Tail queue access methods.
*/
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head)
 == STAILQ_END(head))

/*
* Singly-linked Tail queue functions.
*/
#define STAILQ_INIT(head) do {      \
 (head)->stqh_first = NULL;     \
 (head)->stqh_last = &(head)->stqh_first;    \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do {   \
 if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
  (head)->stqh_last = &(elm)->field.stqe_next;  \
 (head)->stqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do {   \
 (elm)->field.stqe_next = NULL;     \
 *(head)->stqh_last = (elm);     \
```

```
 (head)->stqh_last = &(elm)->field.stqe_next;   \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL)\
  (head)->stqh_last = &(elm)->field.stqe_next;  \
 (listelm)->field.stqe_next = (elm);    \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE_HEAD(head, field) do {    \
 if (((head)->stqh_first = (head)->stqh_first->field.stqe_next)
 == NULL) \
  (head)->stqh_last = &(head)->stqh_first;   \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE(head, elm, type, field) do {   \
 if ((head)->stqh_first == (elm)) {    \
  STAILQ_REMOVE_HEAD((head), field);   \
 } else {       \
  struct type *curelm = (head)->stqh_first;  \
  while (curelm->field.stqe_next != (elm))   \
   curelm = curelm->field.stqe_next;  \
  if ((curelm->field.stqe_next =    \
   curelm->field.stqe_next->field.stqe_next) == NULL) \
      (head)->stqh_last = &(curelm)->field.stqe_next; \
 }       \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->stqh_first);    \
  (var);       \
  (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = STAILQ_FIRST((head));    \
    (var) && ((tvar) = STAILQ_NEXT((var), field), 1);  \
    (var) = (tvar))

#define STAILQ_CONCAT(head1, head2) do {    \
 if (!STAILQ_EMPTY((head2))) {     \
  *(head1)->stqh_last
 = (head2)->stqh_first;  \
  (head1)->stqh_last = (head2)->stqh_last;  \
  STAILQ_INIT((head2));     \
 }       \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field)     \
```

```
 (STAILQ_EMPTY((head)) ?     \
  NULL :      \
        ((struct type *)(void *)    \
  ((char *)((head)->stqh_last) - offsetof(struct type, field))))


#ifndef _KERNEL
/*
* Circular queue definitions. Do not use. We still keep the macros
* for compatibility but because of pointer aliasing issues their use
* is discouraged!
*/

/*
* __launder_type():  We use this ugly hack to work around the the compiler
* noticing that two types may not alias each other and elide tests in code.
* We hit this in the CIRCLEQ macros when comparing 'struct name *' and
* 'struct type *' (see CIRCLEQ_HEAD()).  Modern compilers (such as GCC
* 4.8) declare these comparisons as always false, causing the code to
* not run as designed.
*
* This hack is only to be used for comparisons and thus can be fully const.
 * Do not use for assignment.
*
* If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
* this by changing the head/tail sentinal values, but see the note above
* this one.
*/
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
 __asm __volatile("" : "+r" (__x));
 return __x;
}

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)    \
 if ((head)->cqh_first != CIRCLEQ_ENDC(head) &&   \
    (head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
  QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
       __FILE__, __LINE__);    \
 if ((head)->cqh_last != CIRCLEQ_ENDC(head) &&   \
    (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
  QUEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
       __FILE__, __LINE__);
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)   \
 if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) {  \
```

```
 if ((head)->cqh_last != (elm))    \
  QUEUEDEBUG_ABORT("CIRCLEQ
elm last %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 } else {       \
 if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
  QUEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 }       \
 if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) {  \
 if ((head)->cqh_first != (elm))    \
  QUEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 } else {       \
 if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
  QUEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 }
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)   \
 (elm)->field.cqe_next = (void *)1L;    \
 (elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type)     \
struct name {        \
 struct type *cqh_first;  /*
 first element */  \
 struct type *cqh_last;  /* last element */  \
}

#define CIRCLEQ_HEAD_INITIALIZER(head)     \
 { CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type)      \
struct {        \
 struct type *cqe_next;  /* next element */  \
 struct type *cqe_prev;  /* previous element */  \
}

/*
* Circular queue functions.
*/
#define CIRCLEQ_INIT(head) do {      \
 (head)->cqh_first = CIRCLEQ_END(head);    \
```

```
 (head)->cqh_last = CIRCLEQ_END(head);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field)  \
 (elm)->field.cqe_next = (listelm)->field.cqe_next;  \
 (elm)->field.cqe_prev = (listelm);    \
 if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head))  \
  (head)->cqh_last = (elm);    \
 else       \
  (listelm)->field.cqe_next->field.cqe_prev = (elm); \
 (listelm)->field.cqe_next = (elm);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_BEFORE(head,
 listelm, elm, field) do {  \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field)  \
 (elm)->field.cqe_next = (listelm);    \
 (elm)->field.cqe_prev = (listelm)->field.cqe_prev;  \
 if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head))  \
  (head)->cqh_first = (elm);    \
 else       \
  (listelm)->field.cqe_prev->field.cqe_next = (elm); \
 (listelm)->field.cqe_prev = (elm);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do {   \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 (elm)->field.cqe_next = (head)->cqh_first;   \
 (elm)->field.cqe_prev = CIRCLEQ_END(head);   \
 if ((head)->cqh_last == CIRCLEQ_ENDC(head))   \
  (head)->cqh_last = (elm);    \
 else       \
  (head)->cqh_first->field.cqe_prev = (elm);  \
 (head)->cqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do {   \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 (elm)->field.cqe_next
 = CIRCLEQ_END(head);   \
 (elm)->field.cqe_prev = (head)->cqh_last;   \
 if ((head)->cqh_first == CIRCLEQ_ENDC(head))   \
  (head)->cqh_first = (elm);    \
 else       \
  (head)->cqh_last->field.cqe_next = (elm);  \
```

```
 (head)->cqh_last = (elm);     \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_REMOVE(head, elm, field) do {    \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field)   \
 if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head))  \
  (head)->cqh_last = (elm)->field.cqe_prev;  \
 else       \
  (elm)->field.cqe_next->field.cqe_prev =   \
      (elm)->field.cqe_prev;    \
 if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head))  \
  (head)->cqh_first = (elm)->field.cqe_next;  \
 else       \
  (elm)->field.cqe_prev->field.cqe_next =   \
      (elm)->field.cqe_next;    \
 QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->cqh_first);    \
  (var) != CIRCLEQ_ENDC(head);    \
  (var)
 = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field)   \
 for ((var) = ((head)->cqh_last);    \
  (var) != CIRCLEQ_ENDC(head);    \
  (var) = ((var)->field.cqe_prev))

/*
* Circular queue access methods.
*/
#define CIRCLEQ_FIRST(head)  ((head)->cqh_first)
#define CIRCLEQ_LAST(head)  ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head)  (__launder_type(head))
/* For assignments */
#define CIRCLEQ_END(head)  ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head)      \
   (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field)    \
 (((elm)->field.cqe_next == CIRCLEQ_ENDC(head))   \
     ? ((head)->cqh_first)     \
     : (elm->field.cqe_next))
```

```
#define CIRCLEQ_LOOP_PREV(head, elm, field)    \
 (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head))   \
     ? ((head)->cqh_last)     \
     :
 (elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright"                  # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
 gpep517 build-wheel \
  --wheel-dir .dist \
  --output-fd 3 3>&1 >&2
}

package() {
 gpep517 install-wheel --destdir "$pkgdir" \
  .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
  flake8-copyright-0.2.4.tar.gz
"
#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated
```

```
#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef  __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__ ((__nothrow__))
# define __NTH(fct) __attribute__ ((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct)     fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
 default_prepare

 cargo fetch --target="$CTARGET" --locked
}

build() {
 cargo auditable build --release --frozen
}
```

```
check() {
 cargo test --frozen
}

package() {
 install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"
/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
 AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y)      \
 ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) ||   \
  (__GNUC__ > (x)))
```

```
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

/*
* This file defines data structures for different types of trees:
* splay trees and red-black trees.
*
* A splay tree is a self-organizing
 data structure.  Every operation
* on the tree causes a splay to happen.  The splay moves the requested
* node to the root of the tree and partly rebalances it.
*
* This has the benefit that request locality causes faster lookups as
* the requested nodes move to the top of the tree.  On the other hand,
* every lookup causes memory writes.
*
* The Balance Theorem bounds the total access time for m operations
* and n inserts on an initially empty tree as O((m + n)lg n).  The
* amortized cost for a sequence of m accesses to a splay tree is O(lg n);
*
* A red-black tree is a binary search tree with the node color as an
* extra attribute.  It fulfills a set of conditions:
* - every search path from the root to a leaf consists of the
*   same number of black nodes,
* - each red node (except for the root) has a black parent,
* - each leaf node is black.
*
* Every operation on a red-black tree is bounded as O(lg n).
* The maximum height of a red-black tree is 2lg (n+1).
 */

#define SPLAY_HEAD(name, type)      \
struct name {       \
 struct type *sph_root; /* root of the tree */   \
}

#define SPLAY_INITIALIZER(root)     \
 { NULL }

#define SPLAY_INIT(root) do {      \
 (root)->sph_root = NULL;     \
```

```
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type)      \
struct {        \
 struct type *spe_left; /* left element */   \
 struct type *spe_right; /* right element */   \
}

#define SPLAY_LEFT(elm, field)  (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field)  (elm)->field.spe_right
#define SPLAY_ROOT(head)  (head)->sph_root
#define SPLAY_EMPTY(head)  (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do {   \
 SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
 SPLAY_RIGHT(tmp, field) = (head)->sph_root;   \
 (head)->sph_root = tmp;      \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do {   \
 SPLAY_RIGHT((head)->sph_root,
 field) = SPLAY_LEFT(tmp, field); \
 SPLAY_LEFT(tmp, field) = (head)->sph_root;   \
 (head)->sph_root = tmp;      \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do {    \
 SPLAY_LEFT(tmp, field) = (head)->sph_root;   \
 tmp = (head)->sph_root;      \
 (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do {    \
 SPLAY_RIGHT(tmp, field) = (head)->sph_root;   \
 tmp = (head)->sph_root;      \
 (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do {  \
 SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
 SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field);\
 SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
 SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
```

```
#define
 SPLAY_PROTOTYPE(name, type, field, cmp)    \
void name##_SPLAY(struct name *, struct type *);   \
void name##_SPLAY_MINMAX(struct name *, int);    \
struct type *name##_SPLAY_INSERT(struct name *, struct type *);  \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *);  \
        \
/* Finds the node with the same key as elm */    \
static __inline struct type *      \
name##_SPLAY_FIND(struct name *head, struct type *elm)   \
{        \
 if (SPLAY_EMPTY(head))      \
  return(NULL);      \
 name##_SPLAY(head, elm);     \
 if ((cmp)(elm, (head)->sph_root) == 0)    \
  return (head->sph_root);    \
 return (NULL);       \
}        \
        \
static __inline _sys_tree_h_unused struct type *   \
name##_SPLAY_NEXT(struct name *head, struct type *elm)   \
{        \
 name##_SPLAY(head, elm);     \
 if (SPLAY_RIGHT(elm, field) != NULL) {    \
  elm = SPLAY_RIGHT(elm, field);    \
  while (SPLAY_LEFT(elm, field) != NULL) {  \
   elm = SPLAY_LEFT(elm, field);   \
  }       \
 }
 else       \
  elm = NULL;      \
 return (elm);       \
}        \
        \
static _sys_tree_h_unused __inline struct type *   \
name##_SPLAY_MIN_MAX(struct name *head, int val)   \
{        \
 name##_SPLAY_MINMAX(head, val);     \
       return (SPLAY_ROOT(head));     \
}

/* Main splay operation.
* Moves node close to the key of elm to top
*/
#define SPLAY_GENERATE(name, type, field, cmp)    \
struct type *       \
name##_SPLAY_INSERT(struct name *head, struct type *elm)  \
{        \
```

```
  if (SPLAY_EMPTY(head)) {      \
    SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
  } else {        \
    int __comp;       \
    name##_SPLAY(head, elm);     \
    __comp = (cmp)(elm, (head)->sph_root);   \
    if(__comp < 0) {      \
     SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field);\
     SPLAY_RIGHT(elm, field) = (head)->sph_root;  \
     SPLAY_LEFT((head)->sph_root, field) = NULL;  \
    } else if (__comp > 0) {     \
     SPLAY_RIGHT(elm,
 field) = SPLAY_RIGHT((head)->sph_root, field);\
     SPLAY_LEFT(elm, field) = (head)->sph_root;  \
     SPLAY_RIGHT((head)->sph_root, field) = NULL; \
    } else       \
     return ((head)->sph_root);    \
  }         \
  (head)->sph_root = (elm);      \
  return (NULL);       \
}         \
        \
struct type *        \
name##_SPLAY_REMOVE(struct name *head, struct type *elm)  \
{         \
 struct type *__tmp;      \
 if (SPLAY_EMPTY(head))      \
  return (NULL);      \
 name##_SPLAY(head, elm);     \
 if ((cmp)(elm, (head)->sph_root) == 0) {   \
  if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
   (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
  } else {      \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
   name##_SPLAY(head, elm);   \
   SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
  }       \
 return (elm);      \
 }        \
 return (NULL);       \
}         \
        \
void         \
name##_SPLAY(struct
 name *head, struct type *elm)   \
{         \
 struct type __node, *__left, *__right, *__tmp;   \
```

```
 int __comp;      \
\
 SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
 __left = __right = &__node;     \
\
 while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) {  \
  if (__comp < 0) {     \
   __tmp = SPLAY_LEFT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if ((cmp)(elm, __tmp) < 0){   \
    SPLAY_ROTATE_RIGHT(head, __tmp, field); \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
     break;    \
   }      \
   SPLAY_LINKLEFT(head, __right, field);  \
  } else if (__comp > 0) {    \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if ((cmp)(elm, __tmp) > 0){   \
    SPLAY_ROTATE_LEFT(head, __tmp, field); \
    if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
     break;    \
   }      \
   SPLAY_LINKRIGHT(head, __left, field);  \
  }       \
 }       \
 SPLAY_ASSEMBLE(head,
 &__node, __left, __right, field);  \
}        \
        \
/* Splay with either the minimum or the maximum element   \
* Used to find minimum or maximum element in tree.   \
*/        \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{        \
 struct type __node, *__left, *__right, *__tmp;   \
\
 SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
 __left = __right = &__node;     \
\
 while (1) {       \
  if (__comp < 0) {     \
   __tmp = SPLAY_LEFT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if (__comp < 0){    \
```

```
    SPLAY_ROTATE_RIGHT(head, __tmp, field); \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
      break;    \
    }      \
    SPLAY_LINKLEFT(head, __right, field);  \
   } else if (__comp > 0) {    \
    __tmp = SPLAY_RIGHT((head)->sph_root, field); \
    if (__tmp == NULL)    \
     break;     \
    if (__comp > 0) {    \
     SPLAY_ROTATE_LEFT(head, __tmp, field); \
     if (SPLAY_RIGHT((head)->sph_root,
  field) == NULL)\
      break;    \
    }      \
    SPLAY_LINKRIGHT(head, __left, field);  \
   }       \
  }       \
  SPLAY_ASSEMBLE(head, &__node, __left, __right, field);  \
 }

 #define SPLAY_NEGINF -1
 #define SPLAY_INF 1

 #define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
 #define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
 #define SPLAY_FIND(name, x, y)  name##_SPLAY_FIND(x, y)
 #define SPLAY_NEXT(name, x, y)  name##_SPLAY_NEXT(x, y)
 #define SPLAY_MIN(name, x)  (SPLAY_EMPTY(x) ? NULL \
     : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
 #define SPLAY_MAX(name, x)  (SPLAY_EMPTY(x) ? NULL \
     : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

 #define SPLAY_FOREACH(x, name, head)     \
  for ((x) = SPLAY_MIN(name, head);    \
       (x) != NULL;      \
       (x) = SPLAY_NEXT(name, head, x))

 /* Macros that define a red-black tree */
 #define RB_HEAD(name, type)      \
 struct name {        \
  struct type *rbh_root; /* root of the tree */   \
 }

 #define RB_INITIALIZER(root)      \
  {
  NULL }
```

```
#define RB_INIT(root) do {      \
 (root)->rbh_root = NULL;     \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED  1
#define RB_ENTRY(type)       \
struct {        \
 struct type *rbe_left;  /* left element */  \
 struct type *rbe_right;  /* right element */  \
 struct type *rbe_parent; /* parent element */  \
 int rbe_color;   /* node color */  \
}

#define RB_LEFT(elm, field)  (elm)->field.rbe_left
#define RB_RIGHT(elm, field)  (elm)->field.rbe_right
#define RB_PARENT(elm, field)  (elm)->field.rbe_parent
#define RB_COLOR(elm, field)  (elm)->field.rbe_color
#define RB_ROOT(head)   (head)->rbh_root
#define RB_EMPTY(head)   (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do {     \
 RB_PARENT(elm, field) = parent;     \
 RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL;  \
 RB_COLOR(elm, field) = RB_RED;     \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do {    \
 RB_COLOR(black, field) = RB_BLACK;    \
 RB_COLOR(red, field)
 = RB_RED;     \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do {   \
 (tmp) = RB_RIGHT(elm, field);     \
 if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
  RB_PARENT(RB_LEFT(tmp, field), field) = (elm);  \
 }       \
 RB_AUGMENT(elm);      \
 if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
  if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
   RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
  else       \
   RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
```

```
 } else       \
  (head)->rbh_root = (tmp);    \
 RB_LEFT(tmp, field) = (elm);     \
 RB_PARENT(elm, field) = (tmp);     \
 RB_AUGMENT(tmp);      \
 if ((RB_PARENT(tmp, field)))     \
  RB_AUGMENT(RB_PARENT(tmp, field));   \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do {   \
 (tmp) = RB_LEFT(elm, field);     \
 if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field))
 != NULL) { \
  RB_PARENT(RB_RIGHT(tmp, field), field) = (elm);  \
 }       \
 RB_AUGMENT(elm);      \
 if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
  if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
   RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
  else      \
   RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
 } else       \
  (head)->rbh_root = (tmp);    \
 RB_RIGHT(tmp, field) = (elm);     \
 RB_PARENT(elm, field) = (tmp);     \
 RB_AUGMENT(tmp);      \
 if ((RB_PARENT(tmp, field)))     \
  RB_AUGMENT(RB_PARENT(tmp, field));   \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp)    \
 RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp)   \
 RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr)  \
attr void name##_RB_INSERT_COLOR(struct name *, struct
 type *);  \
attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
attr struct type *name##_RB_INSERT(struct name *, struct type *); \
attr struct type *name##_RB_FIND(struct name *, struct type *);  \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *);   \
attr struct type *name##_RB_PREV(struct type *);   \
attr struct type *name##_RB_MINMAX(struct name *, int);   \
        \
```

```
/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp)    \
 RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp)   \
 RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr)  \
attr void        \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm)  \
{        \
 struct type
 *parent, *gparent, *tmp;    \
 while ((parent = RB_PARENT(elm, field)) != NULL &&  \
     RB_COLOR(parent, field) == RB_RED) {    \
  gparent = RB_PARENT(parent, field);   \
  if (parent == RB_LEFT(gparent, field)) {  \
   tmp = RB_RIGHT(gparent, field);   \
   if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent;    \
    continue;    \
   }      \
   if (RB_RIGHT(parent, field) == elm) {  \
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    tmp = parent;    \
    parent = elm;    \
    elm = tmp;    \
   }      \
   RB_SET_BLACKRED(parent, gparent, field); \
   RB_ROTATE_RIGHT(head, gparent, tmp, field); \
  } else {      \
   tmp = RB_LEFT(gparent, field);   \
   if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent;    \
    continue;    \
   }      \
   if (RB_LEFT(parent, field) == elm)
  {  \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent;    \
    parent = elm;    \
    elm = tmp;    \
   }      \
   RB_SET_BLACKRED(parent, gparent, field); \
```

```
  RB_ROTATE_LEFT(head, gparent, tmp, field); \
 }      \
 }      \
 RB_COLOR(head->rbh_root, field) = RB_BLACK;   \
}       \
       \
attr void       \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{       \
 struct type *tmp;     \
 while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
     elm != RB_ROOT(head)) {     \
  if (RB_LEFT(parent, field) == elm) {   \
   tmp = RB_RIGHT(parent, field);   \
   if (RB_COLOR(tmp, field) == RB_RED) {  \
    RB_SET_BLACKRED(tmp, parent, field); \
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    tmp = RB_RIGHT(parent, field);  \
   }     \
   if ((RB_LEFT(tmp, field) == NULL ||  \
       RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
       (RB_RIGHT(tmp, field) == NULL ||  \
       RB_COLOR(RB_RIGHT(tmp,
 field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED;  \
    elm = parent;    \
    parent = RB_PARENT(elm, field);  \
   } else {     \
    if (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
     struct type *oleft;  \
     if ((oleft = RB_LEFT(tmp, field)) \
         != NULL)   \
      RB_COLOR(oleft, field) = RB_BLACK;\
     RB_COLOR(tmp, field) = RB_RED; \
     RB_ROTATE_RIGHT(head, tmp, oleft, field);\
     tmp = RB_RIGHT(parent, field); \
    }     \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_RIGHT(tmp, field))  \
     RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    elm = RB_ROOT(head);   \
    break;     \
   }      \
  } else {      \
   tmp = RB_LEFT(parent, field);   \
```

```
  if (RB_COLOR(tmp, field) == RB_RED) {  \
   RB_SET_BLACKRED(tmp, parent, field); \
   RB_ROTATE_RIGHT(head, parent, tmp, field);\
   tmp
= RB_LEFT(parent, field);  \
  }      \
  if ((RB_LEFT(tmp, field) == NULL ||  \
      RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
      (RB_RIGHT(tmp, field) == NULL ||  \
      RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
   RB_COLOR(tmp, field) = RB_RED;  \
   elm = parent;    \
   parent = RB_PARENT(elm, field);  \
  } else {     \
   if (RB_LEFT(tmp, field) == NULL ||\
       RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
    struct type *oright;  \
    if ((oright = RB_RIGHT(tmp, field)) \
        != NULL)   \
     RB_COLOR(oright, field) = RB_BLACK;\
    RB_COLOR(tmp, field) = RB_RED; \
    RB_ROTATE_LEFT(head, tmp, oright, field);\
    tmp = RB_LEFT(parent, field); \
   }     \
   RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
   RB_COLOR(parent, field) = RB_BLACK; \
   if (RB_LEFT(tmp, field))  \
    RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
   RB_ROTATE_RIGHT(head, parent, tmp, field);\
   elm = RB_ROOT(head);   \
   break;     \
  }      \
 }       \
}        \
if
(elm)       \
 RB_COLOR(elm, field) = RB_BLACK;   \
}        \
       \
attr struct type *       \
name##_RB_REMOVE(struct name *head, struct type *elm)   \
{        \
 struct type *child, *parent, *old = elm;   \
 int color;       \
 if (RB_LEFT(elm, field) == NULL)    \
  child = RB_RIGHT(elm, field);    \
 else if (RB_RIGHT(elm, field) == NULL)    \
  child = RB_LEFT(elm, field);    \
```

```
else {       \
 struct type *left;     \
 elm = RB_RIGHT(elm, field);    \
 while ((left = RB_LEFT(elm, field)) != NULL)  \
  elm = left;     \
 child = RB_RIGHT(elm, field);    \
 parent = RB_PARENT(elm, field);    \
 color = RB_COLOR(elm, field);    \
 if (child)      \
  RB_PARENT(child, field) = parent;  \
 if (parent) {      \
  if (RB_LEFT(parent, field) == elm)  \
   RB_LEFT(parent, field) = child;  \
  else      \
   RB_RIGHT(parent, field) = child; \
  RB_AUGMENT(parent);    \
 } else       \
  RB_ROOT(head)
= child;    \
 if (RB_PARENT(elm, field) == old)   \
  parent = elm;     \
 (elm)->field = (old)->field;    \
 if (RB_PARENT(old, field)) {    \
  if (RB_LEFT(RB_PARENT(old, field), field) == old)\
   RB_LEFT(RB_PARENT(old, field), field) = elm;\
  else      \
   RB_RIGHT(RB_PARENT(old, field), field) = elm;\
  RB_AUGMENT(RB_PARENT(old, field));  \
 } else       \
  RB_ROOT(head) = elm;    \
 RB_PARENT(RB_LEFT(old, field), field) = elm;  \
 if (RB_RIGHT(old, field))    \
  RB_PARENT(RB_RIGHT(old, field), field) = elm; \
 if (parent) {      \
  left = parent;     \
  do {      \
   RB_AUGMENT(left);   \
  } while ((left = RB_PARENT(left, field)) != NULL); \
 }       \
 goto color;      \
}        \
parent = RB_PARENT(elm, field);     \
color = RB_COLOR(elm, field);     \
if (child)       \
 RB_PARENT(child, field) = parent;   \
if (parent) {       \
 if (RB_LEFT(parent, field) == elm)   \
  RB_LEFT(parent, field) = child;   \
```

```
  else      \
   RB_RIGHT(parent,
 field) = child;  \
  RB_AUGMENT(parent);     \
 } else       \
  RB_ROOT(head) = child;     \
color:        \
 if (color == RB_BLACK)      \
  name##_RB_REMOVE_COLOR(head, parent, child);  \
 return (old);       \
}        \
       \
/* Inserts a node into the RB tree */     \
attr struct type *       \
name##_RB_INSERT(struct name *head, struct type *elm)   \
{        \
 struct type *tmp;      \
 struct type *parent = NULL;     \
 int comp = 0;       \
 tmp = RB_ROOT(head);      \
 while (tmp) {       \
  parent = tmp;      \
  comp = (cmp)(elm, parent);    \
  if (comp < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else if (comp > 0)     \
   tmp = RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }       \
 RB_SET(elm, parent, field);     \
 if (parent != NULL) {      \
  if (comp < 0)      \
   RB_LEFT(parent, field) = elm;   \
  else       \
   RB_RIGHT(parent, field) = elm;   \
  RB_AUGMENT(parent);     \
 } else       \
  RB_ROOT(head) = elm;     \
 name##_RB_INSERT_COLOR(head,
 elm);    \
 return (NULL);       \
}        \
       \
/* Finds the node with the same key as elm */    \
attr struct type *       \
name##_RB_FIND(struct name *head, struct type *elm)   \
{        \
```

```
 struct type *tmp = RB_ROOT(head);    \
 int comp;       \
 while (tmp) {       \
  comp = cmp(elm, tmp);     \
  if (comp < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else if (comp > 0)     \
   tmp = RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }        \
 return (NULL);       \
}         \
        \
/* Finds the first node greater than or equal to the search key */ \
attr struct type *       \
name##_RB_NFIND(struct name *head, struct type *elm)   \
{         \
 struct type *tmp = RB_ROOT(head);    \
 struct type *res = NULL;     \
 int comp;       \
 while (tmp) {       \
  comp = cmp(elm, tmp);     \
  if (comp < 0) {      \
   res = tmp;     \
   tmp = RB_LEFT(tmp, field);   \
  }       \
  else if (comp > 0)     \
   tmp =
 RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }        \
 return (res);       \
}         \
        \
/* ARGSUSED */       \
attr struct type *       \
name##_RB_NEXT(struct type *elm)     \
{         \
 if (RB_RIGHT(elm, field)) {     \
  elm = RB_RIGHT(elm, field);    \
  while (RB_LEFT(elm, field))    \
   elm = RB_LEFT(elm, field);   \
 } else {       \
  if (RB_PARENT(elm, field) &&    \
      (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
   elm = RB_PARENT(elm, field);   \
```

```
 else {      \
  while (RB_PARENT(elm, field) &&  \
      (elm == RB_RIGHT(RB_PARENT(elm, field), field)))\
   elm = RB_PARENT(elm, field);  \
  elm = RB_PARENT(elm, field);  \
 }      \
}       \
return (elm);      \
}       \
       \
/* ARGSUSED */       \
attr struct type *      \
name##_RB_PREV(struct type *elm)    \
{       \
 if (RB_LEFT(elm, field)) {    \
  elm = RB_LEFT(elm, field);   \
  while (RB_RIGHT(elm, field))   \
   elm = RB_RIGHT(elm, field);  \
 }
 else {      \
  if (RB_PARENT(elm, field) &&   \
     (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
   elm = RB_PARENT(elm, field);  \
  else {      \
   while (RB_PARENT(elm, field) &&  \
      (elm == RB_LEFT(RB_PARENT(elm, field), field)))\
   elm = RB_PARENT(elm, field);  \
  elm = RB_PARENT(elm, field);  \
 }      \
}       \
return (elm);      \
}       \
       \
attr struct type *      \
name##_RB_MINMAX(struct name *head, int val)   \
{       \
 struct type *tmp = RB_ROOT(head);   \
 struct type *parent = NULL;    \
 while (tmp) {      \
  parent = tmp;     \
  if (val < 0)     \
   tmp = RB_LEFT(tmp, field);  \
  else      \
   tmp = RB_RIGHT(tmp, field);  \
 }       \
 return (parent);     \
}
```

```
#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
 y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x)  name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x)  name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head)     \
 for ((x) = RB_MIN(name, head);     \
      (x) != NULL;      \
      (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y)     \
 for ((x) = (y);       \
     ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y)    \
 for ((x) = RB_MIN(name, head);     \
     ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head)    \
 for ((x) = RB_MAX(name, head);     \
      (x) != NULL;      \
      (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y)    \
 for ((x) = (y);       \
     ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,
 head, y)   \
 for ((x) = RB_MAX(name, head);     \
     ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
      (x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
```

```
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-$_kernver.tar.xz
 revert-broken-uapi.patch
 0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch
```

1.9 github.com/aws/aws-sdk-go-v2/service/ssooidc 1.35.3

1.9.1 Available under license :

```
GetLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
```

```
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
 // This member is required.
 Issuer *types.Issuer

 // License name.
 //
 // This member is required.
 LicenseName *string

 // Product name.
 //
 // This member is required.
```

```
 ProductName *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // Date and time range during which the license is valid, in ISO8601-UTC format.
 //
 // This member is required.
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware

RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever

```
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
```

```
  - Fingerprint
//
//   - IssuerName
//
//   - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
```

```
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
```

```
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

```
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {
```

```
// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
```

```
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
```

```
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
```

```
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
```

```
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
```

```
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
```

```
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
             "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
               "smithy.api#required": {}
             }
           }
         },
         "traits": {
           "smithy.api#output": {}
         }
       },
       "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
         "type": "integer"
       },
       "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
         "type": "structure",
         "members": {
           "message": {
             "target": "smithy.api#String"
           }
         },
         "traits": {
           "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
           "smithy.api#error": "server"
         }

 },
       "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
         "type": "operation",
         "input": {
           "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
         },
         "output": {
           "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
         },
         "errors": [
           {
             "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
           },
           {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
           },
           {
             "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
           },
           {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },

          {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
          "type": "structure",
          "members":
 {
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/DeleteLicenseServerEndpoint"

     },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
```

```
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
         "type": "operation",
         "input": {

          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
              }
            },
```

```
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
               }
            },
            "InstanceUserArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
                }
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },

      "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
         "type": "structure",
         "members": {
            "Subnets": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
               "traits": {
```

```
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
},
```

```
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }

            },
            "FailureMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
              }
            },
            "Products": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                "smithy.api#required": {}
              }
            },
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
 about the identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
```

```
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
               }
            },
            "AssociationDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
               }
            },
            "DisassociationDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
         "type": "list",
         "member": {

     "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
       },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
      },
      {

        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
```

```
                "x-amzn-trace-id",
                "content-length",
                 "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
              ],
              "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
              ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager User Subscriptions",
            "smithy.rules#endpointRuleSet": {
              "version": "1.0",
              "parameters": {

      "Region": {
                  "builtIn": "AWS::Region",
                  "required": false,
                  "documentation": "The AWS region used to dispatch the request.",
                  "type": "string"
                },
                "UseDualStack": {
                  "builtIn": "AWS::UseDualStack",
                  "required": true,
                  "default": false,
                  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                  "type": "boolean"
                },
                "UseFIPS": {
                  "builtIn": "AWS::UseFIPS",
                  "required": true,
                  "default": false,
                  "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                  "type": "boolean"
                },
                "Endpoint": {
                  "builtIn": "SDK::Endpoint",
```

```
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
          }
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
```

```
            "type": "error"
        },
        {
            "conditions": [],
            "endpoint": {
                "url": {
                    "ref": "Endpoint"
                },

"properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                     "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
```

```
                    "ref": "UseFIPS"
                  },

true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

"fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      }
                    ]
                  },

                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
```

```
                                    }

                                ]
                                    }
                                ],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }

            ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [

        {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
```

```
                                {
                                  "fn": "getAttr",
                                  "argv": [

                            {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                "fn": "booleanEquals",
                "argv": [
                    {
                    "ref": "UseDualStack"
                    },
                    true
                ]
                }
```

```
                ],
                "rules": [
                    {

            "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

             }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
```

```
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ],
          "type": "tree"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [],
      "error": "Invalid Configuration: Missing Region",
      "type": "error"
    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
```

```
            "UseFIPS": true,
                "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {

    "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
 true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
```

```
                "endpoint":
{
                    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                }
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                }
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS":
false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                }
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
              "expect": {

      "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                }
              },
              "params": {
                "Region": "us-isob-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,

    "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
            "type": "structure",
            "members": {
              "ProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                  "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
                }
              },
              "HealthStatus": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                  "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
              },
              "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
              },
              "Ipv6Address": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>"
            }
          },
          "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
            }
          },
          "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
          },
          "LicenseServerEndpointArn": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
          },
          "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
          },
          "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
          },
          "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
              "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
```

```
            {

            "name": "PROVISIONING",
               "value": "PROVISIONING",
               "documentation": ""
            },
            {
               "name": "PROVISIONING_FAILED",
               "value": "PROVISIONING_FAILED",
               "documentation": ""
            },
            {
               "name": "PROVISIONED",
               "value": "PROVISIONED",
               "documentation": ""
            },
            {
               "name": "DELETING",
               "value": "DELETING",
               "documentation": ""
            },
            {
               "name": "DELETION_FAILED",
               "value": "DELETION_FAILED",
               "documentation": ""
            },
            {
               "name": "DELETED",

"value": "DELETED",
               "documentation": ""
            }
         ]
      }
   },
   "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
         "smithy.api#enum": [
            {
               "name": "HEALTHY",
               "value": "HEALTHY",
               "documentation": ""
            },
            {
               "name": "UNHEALTHY",
               "value": "UNHEALTHY",
               "documentation": ""
```

```
                },
                {
                    "name": "NOT_APPLICABLE",
                    "value": "NOT_APPLICABLE",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                    "smithy.api#required": {}
                }

    }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
               "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
```

```
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                  "smithy.api#required": {}
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
```

```
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
          "type": "structure",

         "members": {
            "InstanceSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/ListLicenseServerEndpoints"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "LicenseServerEndpoints"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
```

```
  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
 {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
```

empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects. "
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " Lists the user-based subscription products available from an identity provider. ",

          "smithy.api#http": {
            "code": 200,
            "method": "POST",

```
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}

 }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
```

```
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }

 },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
              }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output":
 {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
              }
            },
            "SubscriptionStartDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
              }
            },
            "SubscriptionEndDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }

         },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
```

```
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
              "smithy.api#length": {
                "min": 1
              }
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
          "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

       "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }

    },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
          }
        }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
```

```
            "smithy.api#http": {
               "code": 200,
               "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

               }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
            },
            "IdentityProvider":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }

     }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
```

```
        },
        "traits": {
          "smithy.api#length": {

      "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
```

```
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Tags": {
          "type": "map",
          "key": {
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
```

```
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
```

```
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
              "method": "POST",
              "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
          }
        },


       "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
          }
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
```

```
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
```

```
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
```

```
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
```

```
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
```

```
      uses: actions/checkout@v2
      with:
        path: sdkbase
        ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string
```

```
 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
```

```
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
```

```
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
```

```
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string
```

```
// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
```

```
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
```

```
err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
```

```
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
```

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/smithy-go/middleware"

```
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
```

```
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
    ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
             }
           }
         },
         "traits": {
           "smithy.api#input": {}
         }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
         "type": "structure",
         "members": {},
         "traits": {
           "smithy.api#output": {}
         }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
         "type": "structure",
         "members": {
           "Name": {
             "target": "smithy.api#String",
             "traits": {
               "smithy.api#documentation": "<p>The type of name to filter by.</p>"
              }
           },
           "Values": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
               "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
             }
           },
           "Operator": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
             "traits": {
               "smithy.api#documentation": "<p>An operator for filtering results.</p>"
             }
           }
         },
         "traits": {
           "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
         }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
         "type":
```

```
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
           },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"

          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
```

```
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                 "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
            "AmiId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
                }
            },
            "InstanceID": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
        "Region": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
```

```
            "target": "smithy.api#String",
             "traits": {
               "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
          "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
            }
          },
          "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
          },
          "SubscriptionProviderUpdateTime": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
          },
          "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

     "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
             },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
```

```
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
             "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
          },
          "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
          }
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

```
                        true
                    ]
                }

        ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]

}
                        ],
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                        "type": "error"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": {
                                "ref": "Endpoint"
                            },
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"

            }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "rules": [
```

```
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {

"fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ],
            "assign": "PartitionResult"
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {

                "ref": "UseFIPS"
                  },
                  true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
```

```
],
                                    "rules": [
                                      {
                                        "conditions": [
                                          {
                                            "fn": "booleanEquals",
                                            "argv": [
                                              true,
                                              {
                                                "fn": "getAttr",
                                                "argv": [
                                                  {
                                                    "ref": "PartitionResult"

                                },
                                                  "supportsFIPS"
                                                ]
                                              }
                                            ]
                                          },
                                          {
                                            "fn": "booleanEquals",
                                            "argv": [
                                              true,
                                              {
                                                "fn": "getAttr",
                                                "argv": [

                                                 {
                                                    "ref": "PartitionResult"
                                                  },
                                                  "supportsDualStack"
                                                ]
                                              }
                                            ]
                                          }
                                        ],
                                        "rules": [
                                          {
                                            "conditions": [],

      "rules": [
                                              {
                                                "conditions": [],
                                                "endpoint": {
                                                  "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                  "properties": {},
```

```
                "headers": {}
              },
              "type": "endpoint"
            }
          ],

      "type": "tree"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [],
      "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

      "type": "error"
    }
  ],
  "type": "tree"
},
{

  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {

    "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
```

```
                                        },
                                        true

                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "rules": [
                                        {
                                            "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }
                                    ],
                                    "type": "tree"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS is enabled
but this partition does not support FIPS",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true

                ]
                            }
```

```
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {

                                            "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "rules": [

    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"

                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
```

```
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ],
        "type": "tree"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

}
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

    },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
```

```
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": false
```

```
      }
    },
    {

  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
```

```
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }

      },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },

    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
```

```
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {

      "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "Missing region",

            "expect": {
                    "error": "Invalid Configuration: Missing Region"
                }
            }
        ],
        "version": "1.0"
      }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
```

```
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
            "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "OrganizationIntegration": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
                "traits": {
                    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n
</p>\n            </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n            </li>\n
<li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>InstanceType</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Status</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>UsageOperation</code>\n                </p>\n            </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n                <p>\n                    <code>contains</code>\n                </p>\n            </li>\n            <li>\n
<p>\n                    <code>equals</code>\n                </p>\n            </li>\n            <li>\n
                <p>\n                    <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                     "min": 1,
                     "max": 16384
                  }
               }
            }
         },
         "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
         "type": "structure",
         "members": {
            "Instances": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
               "traits": {
                  "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
               }
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
         },
         "errors": [
            {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
```

```
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
              }
            }
          },

          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
          "type": "structure",
          "members": {
            "RegisteredSubscriptionProviders": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
              "traits": {
                "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
          },
          "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

                 }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
                }
            }
        },
```

```
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
             },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {

"name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
           "traits": {
            "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
            "smithy.api#required": {}
          }
        },
        "SecretArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
```

```
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
            }
          }
         },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          },
          "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"
```

```
            }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

       "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

  "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tagKeys": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
 {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
```

```
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

   "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowUpdate": {
              "target": "smithy.api#Boolean",
              "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
              }
            }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
            }

      },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
```

```
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step

spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

InterceptTransmit

1.10 github.com/aws/aws-sdk-go-v2/service/sso 1.29.8

1.10.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

```
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
```

```
Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
```

```
 // This member is required.
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
```

```
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
```

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger

OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
```

```
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
```

```
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
```

```
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
```

DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {
```

```
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
            "DomainName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
                }
            },
            "DomainIpv4List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainIpv6List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
 {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
             "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }

  },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

       {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
```

```
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
```

```
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

         "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/DeleteLicenseServerEndpoint"

          },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
         "type": "structure",
         "members": {
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                  "smithy.api#required": {}
               }
            },
            "ServerType": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
               "traits": {
                  "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                  "smithy.api#required": {}
```

```
  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
```

```
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
      "type": "operation",
      "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
```

```
from the EC2 instance.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },

    "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

                     "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
                }

            },
            "FailureMessage": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
              "smithy.api#required": {}
            }
```

```
            },
            "LastStatusCheckDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date of the last status check.</p>"
               }
            },
            "StatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
               }
            },
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                  "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
          }
        },
        "AssociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
```

```
            }
          },
          "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
```

```
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
```

```
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {

    "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
```

```
                    {
                      "ref": "Endpoint"
                    }
                  ]

  }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },

          "properties": {},
                    "headers": {}
```

```
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "aws.partition",
                         "argv": [
                            {
                                "ref": "Region"
                            }
                        ],
                        "assign": "PartitionResult"
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },

true
                                ]
                            },
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
```

```
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

    "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    },

                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
```

```
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }

        ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
            "type": "error"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [

    {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "fn": "getAttr",
                    "argv": [

              {
                        "ref": "PartitionResult"
                      },
                      "supportsFIPS"
                    ]
                  },
                  true
```

```
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {

    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
```

```
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]

                }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    }
```

```
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",

            "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
```

```
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

    "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
 true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
```

```
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
```

```
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
```

```
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                  }
                },
                "params": {
                  "Region": "us-iso-east-1",
                  "UseFIPS":
false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
```

```
                "Region": "us-east-1",
                "UseFIPS": false,

        "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
               }
            },
            {
              "documentation": "Missing region",
```

```
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
     }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
```

```
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
```

```
            }
          },
          "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
            }
          },
          "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
              "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
```

```
                "name": "PROVISIONED",
                "value": "PROVISIONED",
                "documentation": ""
              },
              {
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
                "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "IdentityProviderSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
         "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
```

```
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
             }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
```

```
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
         "type": "structure",

       "members": {
            "InstanceSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

      {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
```

```
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
         "type": "structure",
         "members": {
            "LicenseServerEndpoints": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits":
{
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
```

```
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
```

```
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
              "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
```

```
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

 }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output":
```

```
{}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
```

```
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
               }
            },
            "SubscriptionStartDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The start date of a subscription.</p>"
               }
            },
            "SubscriptionEndDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The end date of a subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
         "type": "structure",
         "members": {
            "RdsSalCredentialsProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
          "IdentityProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }

        },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
```

```
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
```

```
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
          "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

       "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
```

```
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
```

```
            "ProductUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                  "smithy.api#required": {}
               }

         }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
         "type": "list",
         "member": {
            "target": "smithy.api#String"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
         "type": "list",
         "member": {
            "target": "smithy.api#String"
         },
         "traits": {
            "smithy.api#length": {

      "min": 0,
               "max": 50
            },
            "smithy.api#sensitive": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
         "type": "operation",
```

```
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
```

```
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
            "type": "structure",
            "members": {},
            "traits": {
                "smithy.api#output": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Tags": {
            "type": "map",
            "key": {
                "target": "smithy.api#String"
            },
            "value": {
                "target": "smithy.api#String"
            },
            "traits": {
                "smithy.api#length": {
                    "min": 0,
                    "max": 50
                },
                "smithy.api#sensitive": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
    {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
    request.</p>",
                "smithy.api#error": "client"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {

           "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

          }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
        "smithy.api#http": {

  "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) of the identity provider to update. "
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": " Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the: <ul> Subnets which you want to add to provision VPC endpoints.   Subnets which you want to remove the VPC endpoints from.   Security group ID which permits traffic to the VPC endpoints. ",
                    "smithy.api#required": {}
                }

            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",

```
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
```

```
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
```

```
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
```

```
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
```

```
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
```

```
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
```

```
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string
```

```
 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
```

```
c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
```

```
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
```

```
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
```

```
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
```

```
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
           "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
```

```
        "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
```

```
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
```

```
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
```

```
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }

        ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
```

```
                {
                  "conditions": [
                    {

              "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
                      "assign": "PartitionResult"
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {

                        "ref": "UseFIPS"
                            },
                            true
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }

],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
```

```
                                        "ref": "PartitionResult"

                                },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [

                                {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],

    "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {

      "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

            "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true

                  ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [
                      {
                        "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled
but this partition does not support FIPS",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
```

```
                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

        {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
  }
 ],
 "type": "tree"
 }
 ]
},
"smithy.rules#endpointTests": {
 "testCases": [
  {
   "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
   "expect": {
    "endpoint": {
     "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

 }
   },
   "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
   }
  },
  {
   "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
   "expect": {
    "endpoint": {
     "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
   },
   "params": {
    "Region": "us-east-1",
```

```
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                }

    },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
```

```
"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                      }
                     },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
              "Region": "us-isob-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",

                  "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
            "type": "string",
            "traits": {
              "smithy.api#enum": [
                {
                  "name": "Enabled",
                  "value": "Enabled",
                  "documentation": "Enabled LinuxSubscriptionsDiscovery"
                },
                {
                  "name": "Disabled",
                  "value": "Disabled",
                  "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
              ]
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
            "type": "structure",
            "members": {
              "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                  "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
```

```
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
```

```
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
```

```
"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                    "min": 1,
                    "max": 16384
```

```
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
 return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
 nextToken from a previously truncated response.</p>"
              }
            }
          },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
```

```
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
          "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
                }
            },
            "SubscriptionProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
```

```
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
               },
              {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
              },
              {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
              },
              {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
              }
            ]
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {

      "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
```

```
            },
            "InstanceCount": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
               "traits": {
                  "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
         "type": "list",

       "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "RedHat",
                  "value": "RedHat",
                  "documentation": "RedHat subscription provider namespace"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
         "type":
 "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
```

```
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
 {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
```

```
Region.</p>"
                }
            }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }

    },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
```

```
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.11 goautoneg 0.0.0-20191010083416-a7dc8b61c822

1.11.1 Available under license :

Copyright (c) 2011, Open Knowledge Foundation Ltd.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Open Knowledge Foundation Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.12 smithy-go 1.23.1

1.12.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

1.13 aws-sdk-go-v2-internal-v4a 1.4.11

1.13.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

```
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {
```

```
// License beneficiary.
//
// This member is required.
Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string
```

```
 // Date and time range during which the license is valid, in ISO8601-UTC format.
 //
 // This member is required.
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
```

```
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
```

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
```

```
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
```

```
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
```

```
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
```

```
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
```

```
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
```

```
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
```

```
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
```

```
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
            }
```

```
            },
            "DomainNetworkSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "SELF_MANAGED",
                   "value": "SELF_MANAGED",
                  "documentation": "SELF_MANAGED type of Active Directory"
               },
               {
                  "name": "AWS_MANAGED",
                  "value": "AWS_MANAGED",
                  "documentation": "AWS_MANAGED type of Active Directory"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
         },
         "output": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
         },
         "errors": [
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                  "smithy.api#required": {}
               }
```

```
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
               }
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
          "smithy.api#error": "server"
        }

    },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/CreateLicenseServerEndpoint"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
 {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
      },
      "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}

    }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

     },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/DeregisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
            "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
            },
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Value of the filter.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
            "ActiveDirectoryIdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

                     "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
                }
```

```
            },
            "FailureMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
               }
            },
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                  "smithy.api#required": {}
               }
            },
            "Products": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>A list of provided user-based subscription
```

```
products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
            },
            "OwnerAccountId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                 "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name
```

```
 from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
                }
            },
```

```
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
```

```
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
```

```
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
```

```
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
```

```
                },

            "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "aws.partition",
                        "argv": [
                            {
                                "ref": "Region"
                            }
                        ],
                        "assign": "PartitionResult"
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },

true
                                ]
                            },
                            {
```

```
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

"fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      }
                    ]
                  },

                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }

              ]
                  }
                ],
                "rules": [
                  {
```

```
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

            ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [

            {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "fn": "getAttr",
                                                "argv": [

                                {
                                                        "ref": "PartitionResult"
                                                    },
```

```
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {

    "conditions": [
                        {
                            "fn": "booleanEquals",
```

```
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]

                }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
```

```
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",

        "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
```

```
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS":
 true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
```

```
"endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-gov-east-1",
                "UseFIPS":
false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-gov-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint":
{
                    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                }
            },
            "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": true,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

      "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
```

```
                "url": "https://example.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,

      "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
```

```
                    }
                },
                {
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
```

```
server.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
            "type": "structure",
            "members": {
                "IdentityProviderArn": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                    }
                },
                "ServerType": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                    "traits": {
                        "smithy.api#documentation": "<p>The type of license server.</p>"
                    }
                },
                "ServerEndpoint": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                    "traits": {
                        "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                    }
                },
                "StatusMessage": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                    }
                },
                "LicenseServerEndpointId": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                    }
                },
                "LicenseServerEndpointArn": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                    }
                },
                "LicenseServerEndpointProvisioningStatus": {
                    "target":
```

```
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
```

```
                "value": "PROVISIONING_FAILED",
                "documentation": ""
              },
              {
                "name": "PROVISIONED",
                "value": "PROVISIONED",
                "documentation": ""
              },
              {
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
       "member": {
         "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
          }

  }
      },
      "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
              }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
             }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
```

from a previously truncated response. "
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": " You can use the following filters to streamline results: <ul> Status   InstanceId "
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": " An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects. "
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
          "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
```

```
                }
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
          "type": "structure",
          "members": {
            "LicenseServerEndpoints": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
              "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits":
 {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
```

```
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
```

```
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

},
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
        "smithy.api#http": {
          "method": "GET",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#readonly": {}
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}

 }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
```

```
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
```

```
        "traits": {
          "smithy.api#output":
 {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
```

```
product for this user.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
                }
            },
            "SubscriptionStartDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The start date of a subscription.</p>"
                }
            },
            "SubscriptionEndDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
```

```
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }

         },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
```

```
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"

            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
          "type": "structure",
          "members": {
            "SecretId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                  "min": 1
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
 {
          "type": "string",
```

```
        "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
          "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

       "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
```

```
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Settings": {
          "type": "structure",
          "members": {
            "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }

      },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
          "type": "operation",
```

```
        "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
```

```
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
       },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
         },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
```

```
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
```

```
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {

     "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
```

```
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
```

```
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
        "smithy.api#http": {

  "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "IdentityProviderArn":
 {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
              }
            },
            "UpdateSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
              "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                "smithy.api#required": {}
              }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
```

```
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
```

```
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string
```

```
 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
```

```
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
```

```
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
```

```
 params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string
```

```
 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
```

```
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
```

```
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
```

```
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
```

```
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType
```

```
 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
```

```
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
```

```
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
```

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
           "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
```

```
        "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
```

```
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
```

```
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
```

```
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }

        ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
```

```
                {
                  "conditions": [
                    {

                "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
                      "assign": "PartitionResult"
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {

                       "ref": "UseFIPS"
                            },
                            true
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }

],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
```

```
                                        "ref": "PartitionResult"

                    },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [

                                     {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {

      "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

            "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled
but this partition does not support FIPS",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
```

```
                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

        {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
  }
 ],
 "type": "tree"
 }
 ]
},
"smithy.rules#endpointTests": {
 "testCases": [
  {
   "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
   "expect": {
    "endpoint": {
     "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

 }
   },
   "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
   }
  },
  {
   "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
   "expect": {
    "endpoint": {
     "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
   },
   "params": {
    "Region": "us-east-1",
```

```
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                }

    },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
```

```
"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                      }
                     },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
              "Region": "us-isob-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",

                  "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
            "type": "string",
            "traits": {
              "smithy.api#enum": [
                {
                  "name": "Enabled",
                  "value": "Enabled",
                  "documentation": "Enabled LinuxSubscriptionsDiscovery"
                },
                {
                  "name": "Disabled",
                  "value": "Disabled",
                  "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
              ]
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
            "type": "structure",
            "members": {
              "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                  "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
```

```
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
```

```
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
```

```
"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                    "min": 1,
                    "max": 16384
```

```
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
                  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
 return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
 nextToken from a previously truncated response.</p>"
              }
            }
          },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
```

```
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
          "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
                }
            },
            "SubscriptionProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
```

```
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
               },
              {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
              },
              {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
              },
              {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
              }
            ]
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {

      "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
```

```
            },
            "InstanceCount": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
               "traits": {
                  "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
         "type": "list",

       "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "RedHat",
                  "value": "RedHat",
                  "documentation": "RedHat subscription provider namespace"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
         "type":
 "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
```

```
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
 {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
```

```
Region.</p>"
                }
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
```

```
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.14 libmd 1.1.0-r0

1.14.1 Available under license :

Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/

Files:
*
Copyright:

Copyright 2009, 2011, 2016 Guillem Jover <guillem@hadrons.org>
License: BSD-3-clause
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.
.
THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR
 PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:
include/sha2.h
src/sha2.c
Copyright:
Copyright 2000-2001, Aaron D. Gifford
All rights reserved.
License: BSD-3-clause-Aaron-D-Gifford
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce
 the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of contributors
   may be used to endorse or promote products derived from this software
   without specific prior written permission.
.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTOR(S) ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTOR(S) BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE
 OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Files:
include/rmd160.h
src/rmd160.c
Copyright:
Copyright 2001 Markus Friedl. All rights reserved.
License: BSD-2-clause
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE
 FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:
src/md2.c
Copyright:
Copyright (c) 2001 The NetBSD Foundation, Inc.
All rights reserved.
.
This code is derived from software contributed to The NetBSD Foundation
by Andrew Brown.
License: BSD-2-clause-NetBSD
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions

are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must
 reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
.
THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

Files:
man/rmd160.3
man/sha1.3
man/sha2.3
Copyright:
 Copyright 1997, 2003, 2004 Todd C. Miller <Todd.Miller@courtesan.com>
License: ISC
Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.
.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:
man/mdX.3
src/helper.c
Copyright:
Poul-Henning Kamp <phk@login.dkuug.dk>
License: Beerware
"THE BEER-WARE LICENSE" (Revision 42):
<phk@login.dkuug.dk> wrote this file. As long as you retain

this notice you
can do whatever you want with this stuff. If we meet some day, and you think
this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

Files:
include/md4.h
src/md4.c
Copyright:
Colin Plumb
Todd C. Miller
License: public-domain-md4
This code implements the MD4 message-digest algorithm.
The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.
This code is in the public domain; do with it what you wish.
Todd C. Miller modified the MD5 code to do MD4 based on RFC 1186.

Files:
include/md5.h
src/md5.c
Copyright:
Colin Plumb
License: public-domain-md5
This code implements the MD5 message-digest algorithm.
The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.
This code is in the public domain; do with it what you wish.

Files:
include/sha1.h
src/sha1.c
Copyright:
Steve Reid <steve@edmweb.com>
License: public-domain-sha1
100% Public
Domain

1.15 github.com/baruwa-enterprise/clamd 1.0.1

1.15.1 Available under license :

Mozilla Public License Version 2.0
==================================

1. Definitions

1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of

version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to,

deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method,

process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes
such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes
additional grants as a result of Your choice to
distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this

License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional
Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum
extent possible; and (b)
describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular

Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
```

* ------------------------- *
* *
* Under no circumstances and under no legal theory, whether tort *
* (including negligence), contract, or otherwise, shall any *
* Contributor, or anyone who distributes Covered Software as *
* permitted above, be liable to You for any direct, indirect, *
* special, incidental, or consequential damages of any character *
* including, without limitation,
damages for lost profits, loss of *
* goodwill, work stoppage, computer failure or malfunction, or any *
* and all other commercial damages or losses, even if such party *
* shall have been informed of the possibility of such damages. This *
* limitation of liability shall not apply to liability for death or *
* personal injury resulting from such party's negligence to the *
* extent applicable law prohibits such limitation. Some *
* jurisdictions do not allow the exclusion or limitation of *
* incidental or consequential damages, so this exclusion and *
* limitation may not apply to You. *
* *
****

8. Litigation

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place
of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. Any law or regulation which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2.
Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source
Code Form is subject to the terms of the Mozilla Public
License, v. 2.0. If a copy of the MPL was not distributed with this
file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as
defined by the Mozilla Public License, v. 2.0.

1.16 pax-utils 1.3.8-r1

1.16.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses
are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission
to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program
or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute
verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and

you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program
(or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program
in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled
to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute

software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies
a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED
BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version
2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical
commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

1.17 github.com/aws/aws-sdk-go-v2/credentials 1.18.19

1.17.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

```
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
```

```
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
```

```
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
```

```
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
```

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

```
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
```

```
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
```

```
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
```

```
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
```

```
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
```

```
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
```

```
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
```

RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
```

```
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
      "type": "structure",
      "members": {
        "DomainName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
          }
        },
        "DomainIpv4List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv4
```

```
addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainIpv6List": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
                }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "SELF_MANAGED",
                 "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
              },
              {
```

```
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Arn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
          },
          "output": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
```

```
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
              }
            },
```

```
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
               "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
          }

 },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
          },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
```

```
Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
```

```
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
          "type": "structure",
          "members": {
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}

    }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "LicenseServerEndpoint": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
              "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
```

```
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
               }
            },
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
```

```
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
```

```
                "smithy.api#required": {}
            }
          }
        },

    "traits": {
          "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
```

```
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
```

```
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

}
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
```

```
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [

{
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },

        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
```

```
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {

"fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
```

```
                                    ]
                                }
                            ]
                        },

                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }

                    ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }

            ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [

{
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "fn": "getAttr",
                          "argv": [

                {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {

          "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
```

```
                    "type": "error"

            }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]

         }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
```

```
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
```

```
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",

  "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
```

```
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS":
true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                  }
                },
```

```
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
```

```
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
```

```
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,

         "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
    disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
```

```
                },
                {
                  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                  "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                   }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
          "type": "structure",
          "members": {
            "ProvisioningStatus": {
              "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
              }
            },
```

```
            "HealthStatus": {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
               "traits": {
                  "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
               }
            },
            "Ipv4Address": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
               }
            },
            "Ipv6Address": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
               }
            },
            "ServerType": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
               "traits": {
                  "smithy.api#documentation": "<p>The type of license server.</p>"
               }
            },
            "ServerEndpoint": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
               "traits": {
                  "smithy.api#documentation": "<p>The
```

```
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
            }
          },
          "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code> resource for the RDS license server.</p>"
            }
          },
          "LicenseServerEndpointProvisioningStatus": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
            }
          },
          "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
            }
          },
          "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
              "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
                    "value": "PROVISIONING_FAILED",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONED",
                    "value": "PROVISIONED",
                    "documentation": ""
                },
                {
                    "name": "DELETING",
                    "value": "DELETING",
                    "documentation": ""
                },
                {
                    "name": "DELETION_FAILED",
                    "value": "DELETION_FAILED",
                    "documentation": ""
                },
                {
                    "name": "DELETED",

 "value": "DELETED",
                    "documentation": ""
```

```
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
              "name": "NOT_APPLICABLE",
              "value": "NOT_APPLICABLE",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
        "type": "list",
         "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
```

```
            }

    }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
```

```
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
```

```
            "method": "POST",
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
 from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
 <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
```

contain details about the instances that provide user-based subscriptions and also meet the request criteria. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects. "
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " List the Remote Desktop Services (RDS) License Server endpoints ",

```
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoints": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },

        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
```

```
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

  },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
  you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
```

```
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "InstanceUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }

},
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output":
{}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
 {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

           },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
      "type": "union",

     "members": {
        "RdsSalSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }

    },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
            },
            "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
            },
            "Product": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
```

```
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }

      }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {

   "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
        "smithy.api#http": {
```

```
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
```

```
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
```

```
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
          }
```

```
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
```

```
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
```

```
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan
```

```
on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
```

```
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType
```

```
 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
```

```
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
```

```
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
```

```
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
```

```
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
```

```
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
```

```
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string
```

```
 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

```
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string
```

```
 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
```

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/DeregisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
```

```
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
          "smithy.api#http": {
```

```
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            },
```

```
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
```

```
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {}
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
```

```
data for Linux\n      subscriptions.</p>"
                }
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
            "type": "structure",
            "members": {
              "AmiId": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
                }
              },
              "InstanceID": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
                }
              },
              "InstanceType": {
                "target": "smithy.api#String",
                "traits":
{
                  "smithy.api#documentation": "<p>The instance type of the resource.</p>"
                }
              },
              "AccountID": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
                }
              },
              "Status": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The status of the instance.</p>"
                }
              },
              "Region": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
                }
              },
              "UsageOperation": {
                "target": "smithy.api#String",
```

```
  "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
        "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
        },
        "SubscriptionProviderUpdateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
```

```
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
         "type": "service",
         "version": "2018-05-10",
         "operations": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
       "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
```

```
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
```

```
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }

        ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]

}
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
```

```
                },
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"

  }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {

  "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
```

```
                "ref": "UseFIPS"
                    },
                    true
                ]
                },
                {
                "fn": "booleanEquals",
                "argv": [
                    {
                    "ref": "UseDualStack"
                    },
                    true
                ]
                }

    ],
                "rules": [
                {
                    "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                            {
                                "ref": "PartitionResult"

            },
                                "supportsFIPS"
                            ]
                        }
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [

                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
```

```
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                    "type": "tree"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {

        "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }
```

```
                ],
                "rules": [
                  {
                    "conditions": [
                      {

                "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
                          "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

  "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
                      ],
                      "type": "tree"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled
but this partition does not support FIPS",
                  "type": "error"
```

```
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true

    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {

                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

    {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
```

```
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                    }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
],
"type": "tree"
```

```
      }
    ]
  },
  "smithy.rules#endpointTests": {
    "testCases": [
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

    }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {

     "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
```

```
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
              }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
```

```
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
```

```
        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {

        "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",

          "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
        "version": "1.0"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type": "operation",
```

```
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n          <ul>\n            <li>\n              <p>\n                <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n              <p>\n                <code>AmiID</code>\n              </p>\n            </li>\n
          <li>\n              <p>\n                <code>DualSubscription</code>\n              </p>\n            </li>\n
```

\n \n <code>InstanceID</code>\n \n \n \n \n <code>InstanceType</code>\n \n \n \n \n <code>ProductCode</code>\n \n \n \n \n <code>Region</code>\n \n \n \n \n <code>Status</code>\n \n \n \n \n <code>UsageOperation</code>\n \n \n \n For each filter, you can use one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": " The maximum items to return in a request. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": " An array that contains instance objects. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield

isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
objects. "
}
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
"type": "operation",
"input": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
},
"errors": [
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
},
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
"smithy.api#documentation":
" Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations. ",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
"inputToken": "NextToken",
"outputToken": "NextToken",
"pageSize": "MaxResults",
"items": "Subscriptions"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
"type": "structure",
"members": {
"Filters": {

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n       <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
```

```
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
```

```
the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
```

```
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
          "code": 200,
```

```
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
              "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

              "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
```

\n\t\t\tthis is the status of the offline token. "
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status. "
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": " The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider. "
            }
          }
        },
        "traits": {
          "smithy.api#documentation": " A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider. "
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": " Unable to find the requested Amazon Web Services resource. ",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-

```
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
            },
            {
              "name": "Completed",
              "value": "Completed",
              "documentation": "Completed status"
            },
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
            }
          }
        },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          },
          "smithy.api#uniqueItems": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
```

```
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
{}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
```

```
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
{}
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowUpdate": {
              "target": "smithy.api#Boolean",
              "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
              }
            }

          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
```

```
from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
              "message": {
                  "target": "smithy.api#String"
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
              "smithy.api#error": "client"
          }
      }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.18 libcap 2.76-r0

1.18.1 Available under license :

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise *explicitly* stated, the following text describes the licensed conditions under which the contents of this libcap/cap release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap/cap, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.

2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following

disclaimer in the documentation and/or other materials provided
with the distribution.

3.
The name of any author may not be used to endorse or promote
products derived from this software without their specific prior
written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the
terms of the
GNU General Public License (v2.0 - see below), in which case the
provisions of the GNU GPL are required INSTEAD OF the above
restrictions. (This clause is necessary due to a potential conflict
between the GNU GPL and the restrictions contained in a BSD-style
copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.
This
General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if
you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses,
in effect making the
program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively

when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program
under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you;
rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to

this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed
on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity
of any
such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License
incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to

address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software
Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL
ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it

free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey
the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information
on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which

makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise *explicitly* stated, the following text describes the
licensed conditions under which the contents of this libcap/psx release
may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap/psx, with
or without modification, are permitted provided that the following
conditions are met:

1. Redistributions of source code must retain any existing copyright
  notice, and this entire permission notice in its entirety,
  including the disclaimer of warranties.

2. Redistributions in binary form must reproduce all prior and current
  copyright notices, this list of conditions, and the following
  disclaimer in the documentation and/or other materials provided
  with the distribution.

3.
 The name of any author may not be used to endorse or promote
  products derived from this software without their specific prior
  written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,

BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the
GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it

if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if
you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses,
in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities
other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the
Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program
under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,

and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you;
rather, the intent is to
exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary

form) with the major components
(compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However,
nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed
on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then

the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free

Software
Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL
ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey
the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by

the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information
on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which
makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise *explicitly* stated, the following text describes the
licensed conditions under which the contents of this libcap release

may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap, with
or without modification, are permitted provided that the following
conditions are met:

1. Redistributions of source code must retain any existing copyright
notice, and this entire permission notice in its entirety,
including the disclaimer of warranties.

2. Redistributions in binary form must reproduce all prior and current
copyright notices, this list of conditions, and the following
disclaimer in the documentation and/or other materials provided
with the distribution.

3. The name
of any author may not be used to endorse or promote
products derived from this software without their specific prior
written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of
the
GNU General Public License (v2.0 - see below), in which case the

provisions of the GNU GPL are required INSTEAD OF the above restrictions.

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using
it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make
sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and

(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free
use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program
is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of
Section 1
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition,
mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under

the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily
used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If
distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such

parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing

to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING
ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it
starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public
License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
/* SPDX-License-Identifier: BSD-3-Clause OR LGPL-2.0-or-later */

Unless otherwise *explicitly* stated the following text describes the
licensed conditions under which the contents of this module release
may be distributed:

Redistribution and use in source and binary forms of this module, with
or without modification, are permitted provided that the following
conditions are met:

1. Redistributions of source code must retain any existing copyright
notice, and this entire permission notice in its entirety,
including the disclaimer of warranties.

2. Redistributions in binary form must reproduce all prior and current
copyright notices, this list of conditions, and the following
disclaimer in the documentation and/or other materials provided
with the distribution.

3. The name of any author may not be used to endorse or promote
products derived from this software without their specific prior

written
permission.

ALTERNATIVELY, this product may be distributed under the terms of the
GNU Library General Public License, in which case the provisions of
the GNU LGPL are required INSTEAD OF the above restrictions. (This
clause is necessary due to a potential conflict between the GNU LGPL
and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

1.19 github.com/aws/aws-sdk-go-v2/service/internal/s3shared 1.19.11

1.19.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength

```
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
```

```
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
```

```
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
```

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

```
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
  - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string
```

```
 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
 err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
```

```
        },
        "ActiveDirectorySettings":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
            }
        },
        "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
        },
        "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
            }
          }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
      "type": "structure",
      "members": {
        "DomainName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
          }
        },
        "DomainIpv4List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
            }
```

```
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
```

charges in the Amazon Web Services Billing User Guide. </note>",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/user/AssociateUser"
},
"smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
"type": "structure",
"members": {
"Username": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The user name from the identity provider. ",
"smithy.api#required": {}
}
},
"InstanceId": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The ID of the EC2 instance that provides the user-based
subscription. ",
"smithy.api#required": {}
}
},
"IdentityProvider": {
"target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
"traits": {
"smithy.api#documentation": " The identity provider for the user. ",

"smithy.api#required": {}
}
},
"Domain": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The domain name of the Active Directory that contains
information for the user to associate. "
}
},
"Tags": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
"traits": {
"smithy.api#documentation": " The tags that apply for the user association. "
}

```
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
               "traits": {
                 "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                 "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
          }

 },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
          },
          "errors": [
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/CreateLicenseServerEndpoint"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseServerSettings": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager secret that enables administrators to add or remove users associated with the license server.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code> resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
```

```
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"
```

```
        },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
```

```
    },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies
```

```
details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
```

```
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
           "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
           {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
           },
           {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
           },
           {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
           },
           {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
           },
           {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
           },
           {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
           },
           {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
           }
        ],
        "traits": {
           "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
           "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
           },
           "smithy.api#idempotent": {}
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
           "Username": {
              "target": "smithy.api#String",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
```

```
    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

                 "smithy.api#required": {}
              }
            },
```

```
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }

            },
            "FailureMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
    {
          "type": "structure",
          "members": {
```

```
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        },
        "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                "smithy.api#required": {}
            }
        },
        "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
        },
        "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
        }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
          },
          "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
          },
          "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
             },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
```

```
            "aws.auth#sigv4": {
               "name": "license-manager-user-subscriptions"
            },
            "aws.protocols#restJson1": {},
            "smithy.api#cors": {
               "additionalAllowedHeaders": [
                  "*",
                  "content-type",
                  "x-amz-content-sha256",
                  "x-amz-user-agent",
                  "x-amzn-platform-id",
                  "x-amzn-trace-id",
                  "content-length",
                   "x-api-key",
                  "authorization",
                  "x-amz-date",
                  "x-amz-security-token",
                  "Access-Control-Allow-Headers",
                  "Access-Control-Allow-Methods",
                  "Access-Control-Allow-Origin"
               ],
               "additionalExposedHeaders": [
                  "x-amzn-errortype",
                  "x-amzn-requestid",
                  "x-amzn-trace-id"
               ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager User Subscriptions",
            "smithy.rules#endpointRuleSet": {
               "version": "1.0",
               "parameters": {

      "Region": {
                     "builtIn": "AWS::Region",
                     "required": false,
                     "documentation": "The AWS region used to dispatch the request.",
                     "type": "string"
                  },
                  "UseDualStack": {
                     "builtIn": "AWS::UseDualStack",
                     "required": true,
                     "default": false,
                     "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                     "type": "boolean"
                  },
```

```
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]

    }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }
                  ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [
```

```
{
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },

                "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "aws.partition",
                         "argv": [
                          {
                            "ref": "Region"
                          }
```

```
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

true

```
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
```

"fn": "booleanEquals",

```
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                      ]
                    }

                ]
              }
            ],
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }

        ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
            "type": "error"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
```

```
{
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

            {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"

  }
        ],
```

```
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

         }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
```

```
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",

      "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
```

```
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {

"endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
```

```
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
```

```
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
              "expect": {

        "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,

         "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
```

```
                    "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                     }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
            "type": "structure",
            "members": {
              "ProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                  "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
                }
              },
              "HealthStatus": {
                "target":
```

```
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
         }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
```

```
            "StatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
               }
            },
            "LicenseServerEndpointId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
               }
            },
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
               }
            },
            "LicenseServerEndpointProvisioningStatus": {
               "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
               "traits": {
                  "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
               }
            },
            "LicenseServers": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
               }
            },
            "CreationTime": {
               "target": "smithy.api#Timestamp",

    "traits": {
                  "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
         }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",

"value": "DELETED",
            "documentation": ""
          }
        ]
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
       "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
          }
```

```
  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
```

```
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
```

```
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
```

```
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints </p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
```

```
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
           "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
```

```
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "InstanceUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

      "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
{}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }

          },
            "Settings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
               "traits": {
                  "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
```

```
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
```

```
        "type": "union",

      "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
```

```
                    "min": 1
                },
                "smithy.api#required": {}
            }

    },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
```

```
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
```

```
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) of the product user. "
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": " The domain name of the Active Directory that contains the user for whom to stop the product subscription. "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": " Metadata that describes the start product subscription operation. ",
                    "smithy.api#required": {}
                }

    }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {

```
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {

        "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
```

```
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
          "smithy.api#http": {

   "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
```

```
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
```

```
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
```

```
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
```

```
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]
```

```
permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
     - name: Run Scan code on pr ref
       run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
```

```
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
```

```
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
```

```
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
```

```
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
```

```
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
```

```
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
```

```
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
```

```
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

```
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}
```

```
type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
```

```
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
```

```
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
                 "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
               }
            },
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
                "traits": {
                    "smithy.api#documentation": "<p>An operator for filtering results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
                }
            },
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
                }
            },
            "LastSuccessfulDataRetrievalTime": {
                "target": "smithy.api#String",
                "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/subscription/GetServiceSettings"

          },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {}
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
              }
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
```

subscriptions-usage-operation.html\">Usage\n operation values in the License Manager User Guide. "
}
},
"ProductCode": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
"traits": {
"smithy.api#documentation": " The product code for the instance. For more information, see Usage\n operation values in the License Manager User Guide . "
}
},
"LastUpdatedTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The time in which the last discovery updated the instance details. "
}
},
"SubscriptionName": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the license subscription that the instance uses. "
}
},
"OsVersion": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The operating system software version that runs on your instance. "
}
},
"SubscriptionProviderCreateTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses. "
}
},
"SubscriptionProviderUpdateTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider. "
}
},
"DualSubscription": {
"target": "smithy.api#String",

```
            "traits": {
                "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
        },
        "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
```

```
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
       "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
```

```
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
```

```
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": {
                          "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
```

```
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

                                "ref": "UseFIPS"
                                        },
```

```
                true
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }

  ],
                "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"

        },
                            "supportsFIPS"
                        ]
                      }
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [

                         {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                      }
                    ]
```

```
                }
            ],
            "rules": [
                {
                    "conditions": [],

        "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],

                "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        }
    ],
    "type": "tree"
},
{

    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
```

```
                "conditions": [
                    {

            "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true

                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled
 but this partition does not support FIPS",
                "type": "error"
            }
        ],
        "type": "tree"
```

```
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true

            ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {

                                  "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                              ]
                            }
                          ]
                        }
                      ],
                      "rules": [
                        {
                          "conditions": [],
                          "rules": [

    {
                                "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
```

```
                                    }
                                ],
                                "type": "tree"

                        }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
```

```
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

  }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
```

```
                }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
```

```
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
```

```
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
                },
                {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
                      }
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                },

          {
                    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                    "expect": {
```

```
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {

    "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "Missing region",

        "expect": {
                "error": "Invalid Configuration: Missing Region"
            }
        }
      ],
      "version": "1.0"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
 {
        "type": "structure",
        "members": {
          "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n          <ul>\n            <li>\n              <p>\n                <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n              <p>\n                <code>AmiID</code>\n              </p>\n            </li>\n
          <li>\n              <p>\n                <code>DualSubscription</code>\n              </p>\n            </li>\n
 <li>\n              <p>\n                <code>InstanceID</code>\n              </p>\n            </li>\n            <li>\n
  <p>\n                <code>InstanceType</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
    <code>ProductCode</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
 <code>Region</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
```

<code>Status</code>\n \n \n \n \n <code>UsageOperation</code>\n \n \n \n For each filter, you can use one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
}
},
"MaxResults": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
"traits": {
"smithy.api#documentation": " The maximum items to return in a request. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
"smithy.api#length": {
"min": 1,
"max": 16384
}
}
}
},
"traits": {
"smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
"type": "structure",
"members": {
"Instances": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
"traits": {
"smithy.api#documentation": " An array that contains instance objects. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
objects. "
}

```
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

              "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
```

```
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
match a certain subscription's name.</p>\n          <p>The valid names for this filter are:</p>\n          <ul>\n
<li>\n              <p>\n                  <code>Subscription</code>\n              </p>\n          </li>\n          </ul>\n
<p>The valid Operators for this filter are:</p>\n          <ul>\n          <li>\n              <p>\n
<code>contains</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
<code>equals</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
        <code>Notequal</code>\n              </p>\n          </li>\n          </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
```

```
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",
```

```
    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
        "smithy.api#http": {
          "method": "GET",
          "uri": "/tags/{resourceArn}"
        },
        "smithy.api#readonly": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
```

```
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
             },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
```

```
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
              },
              {

        "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
               "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
               "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

             "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
               "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
```

```
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
                }
            },
             "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
                }
            },
```

```
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
                }
            },
            "LastSuccessfulDataRetrievalTime": {
                "target": "smithy.api#String",
                "traits": {

                    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
```

```
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
```

```
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
 {
        "type": "list",
        "member": {
```

```
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
```

```
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
          "type": "map",
          "key":
{
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
```

```
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

  "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tagKeys": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
```

```
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
 {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required":
{}
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowUpdate": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
          }
        }

      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }
```

```
    },
            "Status": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
               }
            },
            "StatusMessage": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
               "traits": {
                  "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
               }
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
             }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.20 perks 1.0.1

1.20.1 Available under license :

Copyright (C) 2013 Blake Mizerany

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.21 aws-sdk-go-v2-service-internal-accept-encoding 1.13.2

1.21.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

```
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
```

This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

```
 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint

```
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
```

```
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
 spanInitializeStart
 InterceptExecution
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 spanInitializeEnd
 Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
```

```
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
```

```
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
```

OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
"smithy": "2.0",
"shapes": {

```
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

       "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                     "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
 {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
               }
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
         "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
         "type": "structure",
         "members": {
            "message": {
```

```
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
        }

},
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/license-server/CreateLicenseServerEndpoint"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
```

```
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

   "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },

        "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
          "type": "structure",
          "members": {
            "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members":
```

```
{
            "Attribute": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
               }
            },
            "Operation": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
               }
            },
            "Value": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Value of the filter.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
         "type": "union",
         "members": {
            "ActiveDirectoryIdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
         }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
```

```
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                  "smithy.api#required": {}
               }
            },
            "Products": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                  "smithy.api#required": {}
               }
            },
            "LastStatusCheckDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date of the last status check.</p>"
```

```
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
           }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
          }
        },
```

```
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
```

```
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
```

```
 }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [

{
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },

              "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
```

```
    },
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
               "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
```

true

```
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
```

```
                "rules": [
                  {
                    "conditions": [
                      {

        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },

                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsDualStack"
                            ]
                          }

                ]
                      }
                    ],
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
```

```
        ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [

{
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

{
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
```

```
                "conditions": [],
                "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"

}
          ],
          "type": "tree"
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ],
          "rules": [
            {

    "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsDualStack"
```

```
                                    ]

                }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
```

```
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",

            "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
```

```
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                    }
                  },
                  "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                  },
                  "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                  "expect": {

 "endpoint": {
                      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
```

```
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
```

```
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
```

```
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

  "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

   "UseDualStack": false,
        "Endpoint": "https://example.com"
```

```
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
       }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
```

```
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
          },
          "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
              "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
          },
          "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
          },
          "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
              }
            },
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>"
              }
            },
            "ServerEndpoint": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
              }
            },
            "LicenseServerEndpointId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
              }
            },
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code> resource for the RDS license server.</p>"
              }
            },
            "LicenseServerEndpointProvisioningStatus": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
              }
            },
            "LicenseServers": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
              "traits": {
```

```
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
                    "value": "PROVISIONING_FAILED",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONED",
                    "value": "PROVISIONED",
                    "documentation": ""
                },
                {
```

```
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
```

```
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

     "members": {
            "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
     "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
         "type": "structure",
         "members": {
            "LicenseServerEndpoints": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits":
 {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
         },
         "errors": [
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
```

```
</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
```

```
you want to retrieve.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

  "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
```

```
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
```

```
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
```

```
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

          },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
          "type": "string",
```

```
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
      "type": "structure",
      "members": {
        "Endpoint": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
      "type": "union",

     "members": {
        "RdsSalSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }

    },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

         {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
```

```
 }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {

"min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
```

```
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
              "method": "POST",
              "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "IdentityProviderArn":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
        },
        "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                "smithy.api#required": {}
            }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
```

```
            "RemoveSubnets": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                  "smithy.api#required": {}
               }
            },
            "SecurityGroupId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
               "traits": {
                  "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
               }
            }
         },

         "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
```

```
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
```

```
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
```

```
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
```

```
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
```

```
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
```

```
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
```

```
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
```

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata
```

```
 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
```

```
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
```

```
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
```

```
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
```

```
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }
```

```
 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
```

```
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
```

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata

```
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if
err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
```

RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },

    {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                 "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
```

```
        "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"

        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
```

```
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
        "Region": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

  "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
```

```
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

     "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
```

```
        "traits": {
            "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
             },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
            }
        ],
        "traits": {
            "aws.api#service": {
                "sdkId": "License Manager Linux Subscriptions",
```

```
        "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id"
        ],
        "additionalExposedHeaders": [
          "x-amzn-errortype",
          "x-amzn-requestid",
          "x-amzn-trace-id"
        ]
      },
      "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
      "smithy.api#title": "AWS License Manager Linux Subscriptions",
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
```

```
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }

],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
```

```
                        },
                        true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
```

```
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {

                  "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }

],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"

},
                          "supportsFIPS"
                        ]
                      }
```

```
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [

                     {
                          "ref": "PartitionResult"
                      },
                      "supportsDualStack"
                    ]
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],

      "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],

            "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

                "type": "error"
              }
            ],
```

```
                "type": "tree"
            },
            {

        "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

                "fn": "booleanEquals",
                            "argv": [
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true

                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                        "headers": {}
```

```
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled
but this partition does not support FIPS",
                    "type": "error"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true

    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {

                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
```

```
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],
                "rules": [

        {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"

                }
                    ],
                    "type": "tree"
                  },
                  {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does not support DualStack",
                    "type": "error"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
```

```
                    }
                  ],
                  "type": "tree"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
            }
          ],
          "type": "tree"
        }
      ]
    },
    "smithy.rules#endpointTests": {
      "testCases": [
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

      }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
```

```
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                  }

        },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {

          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
```

```
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

      },
      "params": {
```

```
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },

    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

  "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
```

```
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "Missing region",

                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
```

```
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
```

```
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n
</p>\n            </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n            </li>\n
<li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>InstanceType</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Status</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>UsageOperation</code>\n                </p>\n            </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n                <p>\n                    <code>contains</code>\n                </p>\n            </li>\n            <li>\n
<p>\n                    <code>equals</code>\n                </p>\n            </li>\n            <li>\n
                <p>\n                    <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
```

```
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "Subscriptions"
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
         "type": "structure",
         "members": {
            "Filters": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n         <p>The valid names for this filter are:</p>\n         <ul>\n            <li>\n               <p>\n                  <code>Subscription</code>\n               </p>\n            </li>\n         </ul>\n         <p>The valid Operators for this filter are:</p>\n         <ul>\n            <li>\n               <p>\n                  <code>contains</code>\n               </p>\n            </li>\n            <li>\n               <p>\n                  <code>equals</code>\n               </p>\n            </li>\n            <li>\n               <p>\n                  <code>Notequal</code>\n               </p>\n            </li>\n         </ul>"
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                     "min": 1,
                     "max": 16384
                  }
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "NextToken
```

```
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
```

```
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
```

```
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "EQUAL",
                    "value": "Equal",
                    "documentation": "Equal operator"
                },
                {
                    "name": "NOT_EQUAL",
                    "value": "NotEqual",
                    "documentation": "Not equal operator"
```

```
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
          "smithy.api#length": {
            "min": 1,
            "max": 20
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {

          "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
```

```
            },

           {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
         "type": "structure",
         "members": {
            "SubscriptionProviderSource": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                  "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                  "smithy.api#required": {}
               }
            },
            "SecretArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
               "traits": {

          "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
               }
            }
         },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
          }
        },
        "SubscriptionProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
 subscription provider \n\t\t\tthat you requested.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
 Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
```

```
            },
            "SecretArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
               }
            },
            "SubscriptionProviderStatus": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
               }
            },
            "SubscriptionProviderStatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
               }
            },
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {

                  "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
         "type": "structure",
```

```
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n        Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n        Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
                }
            },
            "InstanceCount": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
                "traits": {
                    "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

     "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RedHat",
                    "value": "RedHat",
                    "documentation": "RedHat subscription provider namespace"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
```

```
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
               "method": "PUT",
               "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
         "type": "structure",
         "members": {
            "resourceArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                   "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            },
            "tags": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
         "type": "structure",
         "members": {},
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
         "type": "map",
         "key":
{
            "target": "smithy.api#String"
         },
         "value": {
```

```
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
      "type": "structure",

  "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        },
        "tagKeys": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input":
{}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }

    },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
         }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.22 github.com/confluentinc/confluent-kafka-go/v2 2.12.0

1.22.1 Available under license :

LICENSE

---

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill
2023, Confluent Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished
to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

For src/crc32c.c copied (with modifications) from
http://stackoverflow.com/a/17646775/1821055

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
 This software is provided 'as-is',
 without any express or implied
 warranty. In no event will the author be held liable for any damages
 arising from the use of this software.

 Permission is granted to anyone to use this software for any purpose,
 including commercial applications, and to alter it and redistribute it
 freely, subject to the following restrictions:

 1. The origin of this software must not be misrepresented; you must not
    claim that you wrote the original software. If you use this software
    in a product, an acknowledgment in the product documentation would be
    appreciated but is not required.
 2. Altered source versions must be plainly marked as such, and must not be
    misrepresented as being the original software.
 3. This notice may not be removed or altered from any source distribution.

 Mark Adler
 madler@alumni.caltech.edu
*/

LICENSE.fnv1a

---

parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please
 do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR

PERFORMANCE OF THIS SOFTWARE.

By:
  chongo <Landon Curt Noll> /\oo/\
  http://www.isthe.com/chongo/

Share and Enjoy! :-)

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram https://github.com/codahale/hdrhistogram
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License
 (MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH
 THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE

LICENSE.lz4
---
src/rdxxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library
Copyright (c) 2011-2020, Yann Collet

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
 list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
 list of conditions and the following disclaimer in the documentation and/or
 other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.murmur2
---
parts of src/rdmurmur2.c: git@github.com:abrandoned/murmur2.git

MurMurHash2 Library
//---
// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person
 obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.nanopb

For
files in src/nanopb :
https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or
implied warranty. In no event will the authors be held liable
for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you
must not claim that you wrote the original software. If you use
this software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may
not be removed or altered from any source
distribution.

LICENSE.opentelemetry

For files in src/opentelemetry: https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes
of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated
by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution
incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE
text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding
such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event
and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent
with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.

==========

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted,
free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.queue

For
sys/queue.h:

* Copyright (c) 1991, 1993
* The Regents of the University of California. All rights reserved.
*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 4. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR
 IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* $FreeBSD$

LICENSE.regexp
---
regexp.c and regexp.h from https://github.com/ccxvii/minilibs sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"
These libraries are in the public domain (or the equivalent where that is not
 possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.
"

LICENSE.snappy

##
LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h
originally retrieved from http://github.com/andikleen/snappy-c
git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219
##

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions
 in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING
 NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.tinycthread

From https://github.com/tinycthread/tinycthread/README.txt c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
          2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,

including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from https://github.com/alex85k/wingetopt

```
/*
* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
 ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*-
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
```

* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice,
 this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You"
(or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations,
or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems
that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants
to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work
or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution
of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental,
or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted
against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "{}"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.23 github.com/aws/aws-sdk-go-v2/service/internal/checksum 1.9.2

1.23.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

```
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
```

This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

```
 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint

```
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
```

```
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
 spanInitializeStart
 InterceptExecution
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 spanInitializeEnd
 Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
```

```
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
```

```
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
```

```
 OperationInputValidation
 spanInitializeEnd
Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
```

```
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

       "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                     "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
 {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
               "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
```

```
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
        }

},
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/license-server/CreateLicenseServerEndpoint"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
```

```
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

   "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },

        "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
          "type": "structure",
          "members": {
            "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members":
```

```
{
            "Attribute": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
               }
            },
            "Operation": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
               }
            },
            "Value": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Value of the filter.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
         "type": "union",
         "members": {
            "ActiveDirectoryIdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
         }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

             "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
```

```
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                  "smithy.api#required": {}
               }
            },
            "Products": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                  "smithy.api#required": {}
               }
            },
            "LastStatusCheckDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date of the last status check.</p>"
```

```
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
           }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
          }
        },
```

```
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
```

```
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
```

```
 }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [

{
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },

              "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
```

```
    },
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
               "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
```

true

```
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
```

```
                "rules": [
                  {
                    "conditions": [
                      {

        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },

                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsDualStack"
                            ]
                          }

                ]
                      }
                    ],
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
```

```
        ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [

{
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

            {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
```

```
                                "conditions": [],
                                "endpoint": {

                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled but this partition does not support FIPS",
                        "type": "error"

    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
```

```
                                    ]

                }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
```

```
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",

            "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
```

```
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                    }
                  },
                  "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                  },
                  "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                  "expect": {

 "endpoint": {
                      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
```

```
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
```

```
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
```

```
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

  "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

   "UseDualStack": false,
        "Endpoint": "https://example.com"
```

```
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
       }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
```

```
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
          },
          "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
              "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
          },
          "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
          },
          "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                "smithy.api#documentation": "<p>The <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code> resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
```

```
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
                    "value": "PROVISIONING_FAILED",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONED",
                    "value": "PROVISIONED",
                    "documentation": ""
                },
                {
```

```
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
```

```
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
                "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "IdentityProviderSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
         "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
             }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

     "members": {
            "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
     "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
          "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
         "type": "structure",
         "members": {
            "LicenseServerEndpoints": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits":
 {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
         },
         "errors": [
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
```

```
</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
```

```
you want to retrieve.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

  "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output":
 {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
```

```
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
```

```
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
 reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
 license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
```

```
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

          },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
          "type": "string",
```

```
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
      "type": "structure",
      "members": {
        "Endpoint": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
      "type": "union",

     "members": {
        "RdsSalSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }

    },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

          {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
```

```
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
```

```
 }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {

"min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
```

```
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
              "method": "POST",
              "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "IdentityProviderArn":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
        },
        "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                "smithy.api#required": {}
            }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
```

```
            "RemoveSubnets": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                  "smithy.api#required": {}
               }
            },
            "SecurityGroupId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
               "traits": {
                  "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
               }
            }
         },

         "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
```

```
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
```

```
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
```

```
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
```

```
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
```

```
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
```

```
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
```

```
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
```

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata
```

```
 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
```

```
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
```

```
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
```

```
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
```

```
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }
```

```
 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
```

```
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
```

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata

```
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if
err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
```

RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },

    {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                 "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
```

```
        "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"

        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
```

```
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
        "Region": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

  "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
```

```
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

     "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
```

```
        "traits": {
            "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
             },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
            }
        ],
        "traits": {
            "aws.api#service": {
                "sdkId": "License Manager Linux Subscriptions",
```

```
        "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id"
        ],
        "additionalExposedHeaders": [
          "x-amzn-errortype",
          "x-amzn-requestid",
          "x-amzn-trace-id"
        ]
      },
      "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
      "smithy.api#title": "AWS License Manager Linux Subscriptions",
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
```

```
        "type": "Boolean"
      },
      "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
      }
    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }

],
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
```

```
                        },
                        true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
```

```
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {

                  "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }

],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"

},
                          "supportsFIPS"
                        ]
                      }
```

```
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [

                     {
                          "ref": "PartitionResult"
                      },
                      "supportsDualStack"
                    ]
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],

      "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],

            "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

                "type": "error"
              }
            ],
```

```
                "type": "tree"
            },
            {

        "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

                "fn": "booleanEquals",
                            "argv": [
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true

                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                        "headers": {}
```

```
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS is enabled
but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true

        ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {

                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
```

```
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],
                "rules": [

        {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"

                }
                    ],
                    "type": "tree"
                  },
                  {
                    "conditions": [],
                    "error": "DualStack is enabled but this partition does not support DualStack",
                    "type": "error"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
```

```
                    }
                  ],
                  "type": "tree"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
            }
          ],
          "type": "tree"
        }
      ]
    },
    "smithy.rules#endpointTests": {
      "testCases": [
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

      }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
```

```
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                  }

        },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {

          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
```

```
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

      },
      "params": {
```

```
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
```

```
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },

        {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {

          "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
```

```
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "Missing region",

                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
```

```
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
```

```
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n
</p>\n            </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n            </li>\n
<li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>InstanceType</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Status</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>UsageOperation</code>\n                </p>\n            </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n                <p>\n                    <code>contains</code>\n                </p>\n            </li>\n            <li>\n
<p>\n                    <code>equals</code>\n                </p>\n            </li>\n            <li>\n
                <p>\n                    <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
```

```
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "Subscriptions"
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
         "type": "structure",
         "members": {
            "Filters": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                     "min": 1,
                     "max": 16384
                  }
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "NextToken
```

```
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
```

```
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
```

```
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "EQUAL",
                    "value": "Equal",
                    "documentation": "Equal operator"
                },
                {
                    "name": "NOT_EQUAL",
                    "value": "NotEqual",
                    "documentation": "Not equal operator"
```

```
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
          "smithy.api#length": {
            "min": 1,
            "max": 20
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {

"name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
```

```
            },

           {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
         "type": "structure",
         "members": {
            "SubscriptionProviderSource": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                  "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                  "smithy.api#required": {}
               }
            },
            "SecretArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
               "traits": {

             "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
               }
            }
         },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
                }
            },
            "SubscriptionProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
 subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
 Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
```

```
            },
            "SecretArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
               }
            },
            "SubscriptionProviderStatus": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
               }
            },
            "SubscriptionProviderStatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
               }
            },
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {

                  "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
         "type": "structure",
```

```
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with
Amazon EC2, Bring Your Own\n        Subscription model (BYOS), or from the Amazon Web Services Marketplace.
Certain subscriptions may use licensing from the\n        Amazon Web Services Marketplace as well as OS licensing
from Amazon EC2 or BYOS.</p>"
                }
            },
            "InstanceCount": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
                "traits": {
                    "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

     "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RedHat",
                    "value": "RedHat",
                    "documentation": "RedHat subscription provider namespace"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
```

```
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
               "method": "PUT",
               "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
         "type": "structure",
         "members": {
            "resourceArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                   "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            },
            "tags": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
         "type": "structure",
         "members": {},
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
         "type": "map",
         "key":
{
            "target": "smithy.api#String"
         },
         "value": {
```

```
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
      "type": "structure",

  "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        },
        "tagKeys": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input":
 {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }

    },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
         }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.24 zlib 1.3.1-r2

1.24.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.25 scanelf 1.3.8-r1

1.25.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program
or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute
verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish,
that in
whole or in part contains or is derived from the Program or any

part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can
be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program
(or a work based on it,
under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be

distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program
in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled
to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based
on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the

original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit
royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute
software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies
a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED
BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY
OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if

necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

1.26 github.com/aws/aws-sdk-go-v2/internal/configsources 1.4.11

1.26.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

```
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
 // This member is required.
 Issuer *types.Issuer

 // License name.
 //
 // This member is required.
 LicenseName *string
```

```
 // Product name.
 //
 // This member is required.
 ProductName *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // Date and time range during which the license is valid, in ISO8601-UTC format.
 //
 // This member is required.
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
```

```
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
```

```
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
```

spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming

RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
```

```
//
  - Fingerprint
//
//   - IssuerName
//
//   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
```

```
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {
```

```
// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
```

```
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
```

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
```

```
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

       "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
      "type": "structure",
      "members": {
        "DomainName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
          }
        },
        "DomainIpv4List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
            }
          }
        },
        "DomainIpv6List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
```

```
                }
            }
        },
        "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
        },
        "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "SELF_MANAGED",
                 "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
```

```
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
```

```
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
             "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
               "smithy.api#required": {}
             }
           }
         },
         "traits": {
           "smithy.api#output": {}
         }
       },
       "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
         "type": "integer"
       },
       "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
         "type": "structure",
         "members": {
           "message": {
             "target": "smithy.api#String"
           }
         },
         "traits": {
           "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
           "smithy.api#error": "server"
         }

 },
       "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
         "type": "operation",
         "input": {
           "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
         },
         "output": {
           "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
         },
         "errors": [
           {
             "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
           },
           {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
           },
           {
             "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
           },
           {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
        },

      {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/CreateLicenseServerEndpoint"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
{
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/DeleteLicenseServerEndpoint"

     },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
```

```
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
         "type": "operation",
         "input": {

          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
              }
            },
```

```
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
               }
            },
            "InstanceUserArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
                }
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },

      "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
         "type": "structure",
         "members": {
            "Subnets": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
               "traits": {
```

```
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
},
```

```
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }

            },
            "FailureMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
              }
            },
            "Products": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                "smithy.api#required": {}
              }
            },
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
 about the identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
```

```
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
               }
            },
            "AssociationDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
               }
            },
            "DisassociationDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
         "type": "list",
         "member": {

     "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
       },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
      },
      {

        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
```

```
                "x-amzn-trace-id",
                "content-length",
                 "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
              ],
              "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
              ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager User Subscriptions",
            "smithy.rules#endpointRuleSet": {
              "version": "1.0",
              "parameters": {

      "Region": {
                  "builtIn": "AWS::Region",
                  "required": false,
                  "documentation": "The AWS region used to dispatch the request.",
                  "type": "string"
                },
                "UseDualStack": {
                  "builtIn": "AWS::UseDualStack",
                  "required": true,
                  "default": false,
                  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                  "type": "boolean"
                },
                "UseFIPS": {
                  "builtIn": "AWS::UseFIPS",
                  "required": true,
                  "default": false,
                  "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                  "type": "boolean"
                },
                "Endpoint": {
                  "builtIn": "SDK::Endpoint",
```

```
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
          }
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
```

```
            "type": "error"
        },
        {
            "conditions": [],
            "endpoint": {
                "url": {
                    "ref": "Endpoint"
                },

"properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                     "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
```

```
                    "ref": "UseFIPS"
                  },

true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

"fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      }
                    ]
                  },

                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
```

```
                                    }

                                ]
                                    }
                                ],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }

            ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [

        {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
```

```
                                {
                                  "fn": "getAttr",
                                  "argv": [

                            {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                  ]
                                },
                                true
                              ]
                            }
                          ],
                          "rules": [
                            {
                              "conditions": [],
                              "endpoint": {

                    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                            }
                          ],
                          "type": "tree"
                        },
                        {
                          "conditions": [],
                          "error": "FIPS is enabled but this partition does not support FIPS",
                          "type": "error"

    }
                      ],
                      "type": "tree"
                    },
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }
```

```
                ],
                "rules": [
                    {

            "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

             }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
```

```
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
```

```
            "UseFIPS": true,
                "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {

    "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
 true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
```

```
                "endpoint":
{
                    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                }
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                }
              },
              "params": {
                "Region": "us-iso-east-1",
                "UseFIPS":
false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                }
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
              "expect": {

      "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                }
              },
              "params": {
                "Region": "us-isob-east-1",
```

```
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,

      "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
            "type": "structure",
            "members": {
              "ProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                  "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
                }
              },
              "HealthStatus": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                  "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
              },
              "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
              },
              "Ipv6Address": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>"
            }
          },
          "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
            }
          },
          "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
          },
          "LicenseServerEndpointArn": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
          },
          "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
          },
          "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
          },
          "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
              "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
```

```
            {

            "name": "PROVISIONING",
               "value": "PROVISIONING",
               "documentation": ""
            },
            {
               "name": "PROVISIONING_FAILED",
               "value": "PROVISIONING_FAILED",
               "documentation": ""
            },
            {
               "name": "PROVISIONED",
               "value": "PROVISIONED",
               "documentation": ""
            },
            {
               "name": "DELETING",
               "value": "DELETING",
               "documentation": ""
            },
            {
               "name": "DELETION_FAILED",
               "value": "DELETION_FAILED",
               "documentation": ""
            },
            {
               "name": "DELETED",

"value": "DELETED",
               "documentation": ""
            }
         ]
      }
   },
   "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
         "smithy.api#enum": [
            {
               "name": "HEALTHY",
               "value": "HEALTHY",
               "documentation": ""
            },
            {
               "name": "UNHEALTHY",
               "value": "UNHEALTHY",
               "documentation": ""
```

```
                },
                {
                    "name": "NOT_APPLICABLE",
                    "value": "NOT_APPLICABLE",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                    "smithy.api#required": {}
                }

    }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
```

```
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                  "smithy.api#required": {}
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
```

```
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
          "type": "structure",

         "members": {
            "InstanceSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/ListLicenseServerEndpoints"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "LicenseServerEndpoints"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
```

```
  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
 {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
```

empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects. "
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " Lists the user-based subscription products available from an identity provider. ",

          "smithy.api#http": {
            "code": 200,
            "method": "POST",

```
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
 empty, there are additional elements that the service hasn't included in this request. Use this token with the next
 request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}

 }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
```

```
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }

 },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
              }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output":
 {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {

   "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
              }
            },
            "SubscriptionStartDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
              }
            },
            "SubscriptionEndDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }

          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
```

```
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
              "smithy.api#length": {
                "min": 1
              }
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
          "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

       "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }

    },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
          }
        }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
```

```
            "smithy.api#http": {
               "code": 200,
               "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

               }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
            },
            "IdentityProvider":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }

     }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
```

```
        },
        "traits": {
          "smithy.api#length": {

    "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
```

```
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Tags": {
          "type": "map",
          "key": {
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
```

```
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
```

```
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
              "method": "POST",
              "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
          }
        },


       "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
          }
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
```

```
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
```

```
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
```

```
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
```

```
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
```

```
      uses: actions/checkout@v2
      with:
        path: sdkbase
        ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string
```

```
 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
```

```
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
```

```
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
```

```
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string
```

```
// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
```

```
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
```

```
err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
```

```
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
```

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
```

```
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if
err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
```

```
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
    ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
             }
           }
         },
         "traits": {
           "smithy.api#input": {}
         }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
         "type": "structure",
         "members": {},
         "traits": {
           "smithy.api#output": {}
         }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
         "type": "structure",
         "members": {
           "Name": {
             "target": "smithy.api#String",
             "traits": {
               "smithy.api#documentation": "<p>The type of name to filter by.</p>"
              }
           },
           "Values": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
               "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
             }
           },
           "Operator": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
             "traits": {
               "smithy.api#documentation": "<p>An operator for filtering results.</p>"
             }
           }
         },
         "traits": {
           "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
         }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
         "type":
```

```
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"

          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
```

```
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
          "type": "structure",
          "members": {
            "AmiId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
              }
            },
            "InstanceID": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
        "Region": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
```

```
            "target": "smithy.api#String",
             "traits": {
               "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
          "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
            }
          },
          "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
          },
          "SubscriptionProviderUpdateTime": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
          },
          "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

     "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
             },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
```

```
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
             "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
          },
          "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
          }
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

```
                        true
                    ]
                }

        ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]

}
                        ],
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                        "type": "error"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": {
                                "ref": "Endpoint"
                            },
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"

            }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "rules": [
```

```
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {

"fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ],
            "assign": "PartitionResult"
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {

                "ref": "UseFIPS"
                  },
                  true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
```

```
],
                                    "rules": [
                                      {
                                        "conditions": [
                                          {
                                            "fn": "booleanEquals",
                                            "argv": [
                                              true,
                                              {
                                                "fn": "getAttr",
                                                "argv": [
                                                  {
                                                    "ref": "PartitionResult"

                                },
                                                  "supportsFIPS"
                                                ]
                                              }
                                            ]
                                          },
                                          {
                                            "fn": "booleanEquals",
                                            "argv": [
                                              true,
                                              {
                                                "fn": "getAttr",
                                                "argv": [

                                                 {
                                                    "ref": "PartitionResult"
                                                  },
                                                  "supportsDualStack"
                                                ]
                                              }
                                            ]
                                          }
                                        ],
                                        "rules": [
                                          {
                                            "conditions": [],

      "rules": [
                                              {
                                                "conditions": [],
                                                "endpoint": {
                                                  "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                  "properties": {},
```

```
                "headers": {}
              },
              "type": "endpoint"
            }
          ],

      "type": "tree"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [],
      "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

      "type": "error"
    }
  ],
  "type": "tree"
},
{

  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {

    "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
```

```
                    },
                    true

                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true

  ]
    }
```

```
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {

                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "rules": [

        {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"

                    }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
```

```
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ],
        "type": "tree"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

}
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

    },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
```

```
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": false
```

```
      }
    },
    {

  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
```

```
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
          }

      },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
          }
        },
        "params": {
          "Region": "us-iso-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
          }
        },
        "params": {
          "Region": "us-isob-east-1",
          "UseFIPS": true,
          "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },

    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
```

```
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {

      "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "Missing region",

            "expect": {
                    "error": "Invalid Configuration: Missing Region"
                }
            }
        ],
        "version": "1.0"
      }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
```

```
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
            "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "OrganizationIntegration": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
                "traits": {
                    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n
</p>\n            </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n            </li>\n
<li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>InstanceType</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Status</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>UsageOperation</code>\n                </p>\n            </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n                <p>\n                    <code>contains</code>\n                </p>\n            </li>\n            <li>\n
<p>\n                    <code>equals</code>\n                </p>\n            </li>\n            <li>\n
                <p>\n                    <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                     "min": 1,
                     "max": 16384
                  }
               }
            }
         },
         "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
         "type": "structure",
         "members": {
            "Instances": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
               "traits": {
                  "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
               }
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
         },
         "errors": [
            {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
```

```
                }
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
            }
        }
    },

    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
    "type": "structure",
    "members": {
        "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
                "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
    },
    "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
          ],
          "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

             }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
              }
            }
          },
```

```
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {

"name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
           "traits": {
            "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
            "smithy.api#required": {}
          }
        },
        "SecretArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
```

```
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
            }
          }
         },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          },
          "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"
```

```
            }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n       Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n       Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

       "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

  "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tagKeys": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
 {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {

 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required":
{}
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowUpdate": {
              "target": "smithy.api#Boolean",
              "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
              }
            }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
            }

      },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
```

```
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
            }
         },
         "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
            "type": "structure",
            "members": {
               "message": {
                  "target": "smithy.api#String"
               }
            },
            "traits": {
               "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
               "smithy.api#error": "client"
            }
         }
      }
   }
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step

spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

InterceptTransmit

1.27 github.com/aws/aws-sdk-go-v2/feature/ec2/imds 1.18.11

1.27.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

```
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
```

```
Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
```

```
 // This member is required.
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
```

```
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
```

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger

OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
```

```
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
```

```
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
```

```
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
```

DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {
```

```
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
            "DomainName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
                }
            },
            "DomainIpv4List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainIpv6List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
 {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }

  },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

       {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
```

```
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
```

```
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

         "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/DeleteLicenseServerEndpoint"

          },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
         "type": "structure",
         "members": {
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                  "smithy.api#required": {}
               }
            },
            "ServerType": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
               "traits": {
                  "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                  "smithy.api#required": {}
```

```
  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
```

```
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
```

```
from the EC2 instance.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },

    "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

                     "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
                }

            },
            "FailureMessage": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
              "smithy.api#required": {}
            }
```

```
            },
            "LastStatusCheckDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date of the last status check.</p>"
               }
            },
            "StatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
               }
            },
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                  "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
          }
        },
        "AssociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
```

```
            }
          },
          "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
```

```
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
```

```
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {

    "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
```

```
                {
                  "ref": "Endpoint"
                }
              ]

  }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [

{
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
              "type": "error"
            },
            {
              "conditions": [],
              "endpoint": {
                "url": {
                  "ref": "Endpoint"
                },

        "properties": {},
                  "headers": {}
```

```
            },
            "type": "endpoint"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                 "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },

true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
```

```
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

    "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    },

                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
```

```
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }

        ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
            "type": "error"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [

    {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "fn": "getAttr",
                    "argv": [

              {
                        "ref": "PartitionResult"
                      },
                      "supportsFIPS"
                    ]
                  },
                  true
```

```
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {

    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
```

```
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]

                }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    }
```

```
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",

            "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
```

```
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

    "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
 true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
```

```
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
```

```
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
```

```
                "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                  }
                },
                "params": {
                  "Region": "us-iso-east-1",
                  "UseFIPS":
false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

      "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
```

```
                "Region": "us-east-1",
                "UseFIPS": false,

        "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
               }
            },
            {
              "documentation": "Missing region",
```

```
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
     }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
```

```
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
```

```
            }
          },
          "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
            }
          },
          "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
              "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
```

```
                "name": "PROVISIONED",
                "value": "PROVISIONED",
                "documentation": ""
              },
              {
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
                "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "IdentityProviderSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
         "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
```

```
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
```

```
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
         "type": "structure",

       "members": {
            "InstanceSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

      {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
```

```
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
         "type": "structure",
         "members": {
            "LicenseServerEndpoints": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits":
{
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
```

```
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
```

```
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
              "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
```

```
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

 }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
```

```
{}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
```

```
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
               }
            },
            "SubscriptionStartDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The start date of a subscription.</p>"
               }
            },
            "SubscriptionEndDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The end date of a subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
         "type": "structure",
         "members": {
            "RdsSalCredentialsProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
          "IdentityProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }

        },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
```

```
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
```

```
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
          "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

       "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
```

```
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/StopProductSubscription"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
```

```
            "ProductUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                  "smithy.api#required": {}
               }

         }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
         "type": "list",
         "member": {
            "target": "smithy.api#String"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
         "type": "list",
         "member": {
            "target": "smithy.api#String"
         },
         "traits": {
            "smithy.api#length": {

      "min": 0,
               "max": 50
            },
            "smithy.api#sensitive": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
         "type": "operation",
```

```
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {

           "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

          }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
        "smithy.api#http": {

  "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) of the identity provider to update. "
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": " Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the: <ul> Subnets which you want to add to provision VPC endpoints.   Subnets which you want to remove the VPC endpoints from.   Security group ID which permits traffic to the VPC endpoints. ",
                    "smithy.api#required": {}
                }

            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",

```
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
```

```
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
```

```
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
```

```
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
```

```
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
```

```
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
```

```
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string
```

```
 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
```

```
c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
```

```
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
```

```
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
```

```
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
```

```
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
           "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
           },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
```

```
        "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
```

```
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
 {
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
```

```
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
```

```
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }

        ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
```

```
                {
                  "conditions": [
                    {

              "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
                      "assign": "PartitionResult"
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {

                        "ref": "UseFIPS"
                            },
                            true
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }

],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
```

```
                                        "ref": "PartitionResult"

                    },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [

                                     {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {

      "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

            "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled
but this partition does not support FIPS",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
```

```
                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

        {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
"testCases": [
{
"documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
"expect": {
"endpoint": {
"url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

}
},
"params": {
"Region": "us-east-1",
"UseFIPS": true,
"UseDualStack": true
}
},
{
"documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
"expect": {
"endpoint": {
"url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
}
},
"params": {
"Region": "us-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

    },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
```

```
"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                      }
                     },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
              "Region": "us-isob-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",

                  "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
            "type": "string",
            "traits": {
              "smithy.api#enum": [
                {
                  "name": "Enabled",
                  "value": "Enabled",
                  "documentation": "Enabled LinuxSubscriptionsDiscovery"
                },
                {
                  "name": "Disabled",
                  "value": "Disabled",
                  "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
              ]
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
            "type": "structure",
            "members": {
              "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                  "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
```

```
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
```

```
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
```

```
"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                    "min": 1,
                    "max": 16384
```

```
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

     "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
 return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
 nextToken from a previously truncated response.</p>"
              }
            }
          },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
```

```
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
          "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
                }
            },
            "SubscriptionProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
```

```
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
               },
              {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
              },
              {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
              },
              {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
              }
            ]
```

```
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
                }
```

```
            },
            "InstanceCount": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
               "traits": {
                  "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
         "type": "list",

       "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "RedHat",
                  "value": "RedHat",
                  "documentation": "RedHat subscription provider namespace"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
         "type":
 "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
```

```
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
 {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
```

```
Region.</p>"
                }
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
```

```
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.28 alpine-base 3.22.2-r0

1.28.1 Available under license :

MIT License

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT
 OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
MIT

1.29 alpine-keys 2.5-r0

1.29.1 Available under license :

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
 OR OTHER DEALINGS IN THE SOFTWARE.
Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=alpine-keys
pkgver=2.5
pkgrel=0
pkgdesc="Public keys for Alpine Linux packages"
url="https://alpinelinux.org"
we install arch specific keys to /etc so we cannot do arch=noarch
arch="all"

license="MIT"
replaces="alpine-base"

1.30 logrus 1.9.3

1.30.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF
 OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.31 aws-sdk-go-v2-service-internal-presigned-url 1.13.11

1.31.1 Available under license :

GetLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput

```
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement
```

```
// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string
```

```
// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
```

```
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
```

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
```

```
 return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
  - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
```

```
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
```

```
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
```

```
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
```

```
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
```

```
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
```

AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
```

```
        "members": {
          "DirectoryId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
            "traits": {
              "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
          },
          "ActiveDirectorySettings":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

           "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
```

```
                "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainIpv6List": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
              }
            },
            "DomainNetworkSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "SELF_MANAGED",
                 "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
              },
```

```
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

                "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
                }
```

```
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
         "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
            "smithy.api#error": "server"
         }

 },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
```

```
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

       {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
```

Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details. ",
"smithy.api#required": {}
}
},
"LicenseServerSettings": {
"target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
"traits": {
"smithy.api#documentation": " The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server. ",
"smithy.api#required": {}
}
},
"Tags": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
"traits": {
"smithy.api#documentation": " The tags that apply for the license server endpoint. "
}
}
},
"traits": {
"smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
"type": "structure",
"members": {
"IdentityProviderArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The Amazon Resource Name (ARN) of the identity
provider specified in the request. "
}
},
"LicenseServerEndpointArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The ARN of the <code>LicenseServerEndpoint</code> resource. "
}
}
},
"traits": {
"smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {

```
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
```

```
        "traits": {
          "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/DeleteLicenseServerEndpoint"

          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

    }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
```

```
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

  },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

    },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
```

```
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
```

```
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
          "type": "operation",
          "input": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

   "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
```

```
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
          }
        },

    "traits": {
          "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC
```

endpoints. ",

"smithy.api#required": {}
}
},
"Product": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the user-based subscription product. ",
"smithy.api#required": {}
}
},
"Status": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The status of the identity provider. ",
"smithy.api#required": {}
}
},
"IdentityProviderArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The Amazon Resource Name (ARN) of the identity provider. "
}

},
"FailureMessage": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The failure message associated with an identity provider. "
}
},
"OwnerAccountId": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The AWS Account ID of the owner of this resource. "
}
}
},
"traits": {
"smithy.api#documentation": " Describes an identity provider. "
}
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
"type": "list",
"member": {
"target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
}

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
            },
            "OwnerAccountId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
```

```
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
          }
        },
        "AssociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
          }
        },
        "DisassociationDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {

  "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
```

```
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
```

```
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
```

```
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [

{
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },

        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
```

```
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {

"fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
```

```
                                    ]
                                }
                            ]
                        },

                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }

                    ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }

            ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [

{
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "fn": "getAttr",
                          "argv": [

                {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {

          "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
```

```
                "type": "error"

        }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]

     }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
```

```
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
```

```
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",

  "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
```

```
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS":
true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                  }
                },
```

```
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
```

```
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
```

```
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,

         "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
    disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
```

```
                },
                {
                  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                  "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                   }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
          "type": "structure",
          "members": {
            "ProvisioningStatus": {
              "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
              }
            },
```

```
            "HealthStatus": {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
               "traits": {
                  "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
               }
            },
            "Ipv4Address": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
               }
            },
            "Ipv6Address": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
               }
            },
            "ServerType": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
               "traits": {
                  "smithy.api#documentation": "<p>The type of license server.</p>"
               }
            },
            "ServerEndpoint": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
               "traits": {
                  "smithy.api#documentation": "<p>The
```

```
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
                }
            }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
                    "value": "PROVISIONING_FAILED",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONED",
                    "value": "PROVISIONED",
                    "documentation": ""
                },
                {
                    "name": "DELETING",
                    "value": "DELETING",
                    "documentation": ""
                },
                {
                    "name": "DELETION_FAILED",
                    "value": "DELETION_FAILED",
                    "documentation": ""
                },
                {
                    "name": "DELETED",

 "value": "DELETED",
                    "documentation": ""
```

```
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
              "name": "NOT_APPLICABLE",
              "value": "NOT_APPLICABLE",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
        "type": "list",
         "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
```

```
            }

    }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
```

```
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
```

```
            "method": "POST",
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
 from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
 <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
```

contain details about the instances that provide user-based subscriptions and also meet the request criteria. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects. "
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " List the Remote Desktop Services (RDS) License Server endpoints ",

```
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoints": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },

        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "ProductUserSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
         "type": "structure",
         "members": {
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
```

```
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

  },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
  you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
```

```
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "InstanceUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }

},
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output":
{}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
      "type": "union",

     "members": {
        "RdsSalSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }

    },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
            },
            "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
            },
            "Product": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
```

```
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }

        }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {

   "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
        "smithy.api#http": {
```

```
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
```

```
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
               "method": "POST",
               "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
               }
            },
            "IdentityProviderArn":
 {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
               }
            },
            "UpdateSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
               "traits": {
                  "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
                  "smithy.api#required": {}
```

```
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
          }
```

```
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
```

```
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
```

```
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan
```

```
on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
```

```
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType
```

```
 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
```

```
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
```

```
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
```

```
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
```

```
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
```

```
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
```

```
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string
```

```
 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

```
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string
```

```
 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
```

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/DeregisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
```

```
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
```

```
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
```

```
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                 "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
```

```
data for Linux\n      subscriptions.</p>"
                }
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
            "type": "structure",
            "members": {
              "AmiId": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
                }
              },
              "InstanceID": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
                }
              },
              "InstanceType": {
                "target": "smithy.api#String",
                "traits":
{
                  "smithy.api#documentation": "<p>The instance type of the resource.</p>"
                }
              },
              "AccountID": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
                }
              },
              "Status": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The status of the instance.</p>"
                }
              },
              "Region": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
                }
              },
              "UsageOperation": {
                "target": "smithy.api#String",
```

```
   "traits": {
                    "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
                }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
            },
            "LastUpdatedTime": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
               }
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
```

```
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
         "type": "service",
         "version": "2018-05-10",
         "operations": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
       "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
```

```
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
```

```
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }

        ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]

}
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
```

```
                    },
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
```

```
                "ref": "UseFIPS"
                    },
                    true
                ]
                },
                {
                "fn": "booleanEquals",
                "argv": [
                    {
                    "ref": "UseDualStack"
                    },
                    true
                ]
                }

],
            "rules": [
                {
                "conditions": [
                    {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                        "fn": "getAttr",
                        "argv": [
                            {
                            "ref": "PartitionResult"

},
                            "supportsFIPS"
                        ]
                        }
                    ]
                    },
                    {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                        "fn": "getAttr",
                        "argv": [

                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
```

```
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                    "type": "tree"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {

        "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }
```

```
                ],
                "rules": [
                  {
                    "conditions": [
                      {

                "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
                          "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

  "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
                      ],
                      "type": "tree"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled
but this partition does not support FIPS",
                  "type": "error"
```

```
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true

    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {

                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

    {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
```

```
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                    }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
```

```
      }
    ]
  },
  "smithy.rules#endpointTests": {
    "testCases": [
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

    }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {

     "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
```

```
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
              }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
```

```
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
```

```
        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {

        "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",

          "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
        "version": "1.0"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type": "operation",
```

```
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n          <ul>\n            <li>\n              <p>\n                <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n              <p>\n                <code>AmiID</code>\n              </p>\n            </li>\n
          <li>\n              <p>\n                <code>DualSubscription</code>\n              </p>\n            </li>\n
```

\n \n <code>InstanceID</code>\n \n \n \n \n <code>InstanceType</code>\n \n \n \n \n <code>ProductCode</code>\n \n \n \n \n <code>Region</code>\n \n \n \n \n <code>Status</code>\n \n \n \n \n <code>UsageOperation</code>\n \n \n \n For each filter, you can use one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": " The maximum items to return in a request. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": " An array that contains instance objects. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield

isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
objects. "
}
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
"type": "operation",
"input": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
},
"errors": [
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
},
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
"smithy.api#documentation":
" Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations. ",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
"inputToken": "NextToken",
"outputToken": "NextToken",
"pageSize": "MaxResults",
"items": "Subscriptions"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
"type": "structure",
"members": {
"Filters": {

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
```

```
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
```

```
the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
```

```
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
          "code": 200,
```

```
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
              "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

              "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
```

\n\t\t\tthis is the status of the offline token. "
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status. "
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": " The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider. "
            }
          }
        },
        "traits": {
          "smithy.api#documentation": " A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider. "
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": " Unable to find the requested Amazon Web Services resource. ",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-

```
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
            },
            {
              "name": "Completed",
              "value": "Completed",
              "documentation": "Completed status"
            },
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
            }
          }
        },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          },
          "smithy.api#uniqueItems": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
```

```
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
 {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
```

```
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
{}
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowUpdate": {
              "target": "smithy.api#Boolean",
              "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
              }
            }

          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
```

```
from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
              "message": {
                  "target": "smithy.api#String"
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
              "smithy.api#error": "client"
          }
      }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.32 aws-sdk-go-v2-feature-ec2-imds 1.18.11

1.32.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step

```
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
```

```
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
 // This member is required.
 Issuer *types.Issuer

 // License name.
 //
```

```
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
```

```
= addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
```

```
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
```

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

```
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
```

```
//
//   - Status
//
//
  - Fingerprint
//
//   - IssuerName
//
//   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
```

```
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
```

```
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
```

```
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string
```

```
 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
```

details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration. "
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": " The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory. "
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": " Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false. "
            }
          }
        },
        "traits": {
          "smithy.api#documentation": " Details about an Active Directory identity provider. "
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The domain name for the Active Directory. "
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": " A list of domain IPv4 addresses that are used for the Active Directory. ",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": " A list of domain IPv6 addresses that are used for the Active

```
Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                     "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
```

```
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
           "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }

 },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
```

```
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

          },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) that identifies the identity provider to deregister. "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": " Metadata that describes the results of an identity provider operation. ",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },

```
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based subscriptions.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },

        "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
          "type": "structure",
```

```
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
            "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
            },
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Value of the filter.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

               "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
```

```
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }

            },
            "FailureMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
    {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
    subscriptions.</p>",
                "smithy.api#required": {}
              }
```

```
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                  "smithy.api#required": {}
               }
            },
            "Products": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                  "smithy.api#required": {}
               }
            },
            "LastStatusCheckDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date of the last status check.</p>"
               }
            },
            "StatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
               }
            },
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                  "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
 about the identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
         "type": "list",
```

```
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
 subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {

  "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
         },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
```

```
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                 "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
              ],
              "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
              ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager User Subscriptions",
            "smithy.rules#endpointRuleSet": {
              "version": "1.0",
              "parameters": {

      "Region": {
                  "builtIn": "AWS::Region",
                  "required": false,
                  "documentation": "The AWS region used to dispatch the request.",
                  "type": "string"
                },
                "UseDualStack": {
                  "builtIn": "AWS::UseDualStack",
                  "required": true,
                  "default": false,
                  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                  "type": "boolean"
                },
                "UseFIPS": {
                  "builtIn": "AWS::UseFIPS",
                  "required": true,
                  "default": false,
                  "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
```

```
            "type": "boolean"
          },
          "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
          }
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
```

```
                ]
              }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
          },
          {
            "conditions": [],
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },

"properties": {},
                "headers": {}
            },
            "type": "endpoint"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                 "argv": [
                   {
                     "ref": "Region"
                   }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },

true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

"fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        }
                      ]
                    },

                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
```

```
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }

        ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [

    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [

            {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
                  "type": "error"

}
              ],
              "type": "tree"
            },
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
```

```
                    },
                    true
                ]
            }
        ],
        "rules": [
            {

    "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]

     }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }

                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "DualStack is enabled but this partition does not support DualStack",
                "type": "error"
            }
        ],
        "type": "tree"
```

```
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                     "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
            }
          ]
        },
        "smithy.rules#endpointTests": {
          "testCases": [
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
              },
```

```
        "params": {
          "Region": "us-east-1",

      "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

    "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS":
 true,
          "UseDualStack": true
        }
```

```
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
```

```
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
```

```
                {
                  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint":
{
                      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                  },
                  "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

        "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                    }
```

```
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
```

```
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
          "type": "structure",
          "members": {
            "ProvisioningStatus": {
              "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
              }
            },
            "HealthStatus": {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
              "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
              }
            },
            "Ipv4Address": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
              }
```

```
            },
            "Ipv6Address": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
               }
            },
            "ServerType": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
               "traits": {
                  "smithy.api#documentation": "<p>The type of license server.</p>"
               }
            },
            "ServerEndpoint": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
               "traits": {
                  "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
               }
            },
            "StatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
               }
            },
            "LicenseServerEndpointId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
```

```
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
          },
          "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
          },
          "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
          },
          "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
              "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",

"value": "DELETED",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
```

```
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
              }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
               "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
```

```
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
```

```
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
     },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
         }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
```

```
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
      "type": "structure",

     "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
```

```
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/ListLicenseServerEndpoints"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "LicenseServerEndpoints"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
```

```
  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
```

```
{
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
```

```
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "ProductUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

      },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
               "method": "GET",
               "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
         "type": "structure",
         "members": {
            "ResourceArn": {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                  "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
         "type": "structure",
         "members": {
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}

 }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
```

```
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
        }
    }
},
 "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
```

```
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }

 },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output":
 {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of a product for this user.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
        },
        "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
        },
        "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
```

```
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
                }

         },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
```

```
                "smithy.api#length": {
                    "min": 1
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

 "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
```

```
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
          "type": "operation",
          "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

          {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
```

Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges in the Amazon Web Services Billing User Guide. </note>",
"smithy.api#http": {
"code": 200,
"method": "POST",

"uri": "/user/StartProductSubscription"
}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
"type": "structure",
"members": {
"Username": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The user name from the identity provider of the user. ",
"smithy.api#required": {}
}
},
"IdentityProvider": {
"target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
"traits": {
"smithy.api#documentation": " An object that specifies details for the identity provider. ",
"smithy.api#required": {}
}
},
"Product": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the user-based subscription product. Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> ",
"smithy.api#required": {}
}
},
"Domain": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The domain name of the Active Directory that contains the user for
whom to start the product subscription. "
}
},
"Tags": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
"traits": {
"smithy.api#documentation": " The tags that apply to the product subscription. "

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
                }
            },
            "Domain": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
```

```
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {

    "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
        "smithy.api#http": {
          "method": "PUT",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
```

```
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
 {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

                }
            },
            "TagKeys": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
          "smithy.api#http": {

    "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }

            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
```

```
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
          }
        },

       "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
      }
   }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
```

```
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
```

```
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
```

```
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
```

```
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack,
options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
```

```
      python-version: [3.9]

  steps:
    - name: Checkout target
      uses: actions/checkout@v2
      with:
        path: sdkbase
        ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
```

```
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData
```

```
 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
```

```
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
```

```
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string
```

```
// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string
```

```
// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }
```

```
 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string
```

```
 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
```

```
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
```

```
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
```

```
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
```

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
```

```
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
```

CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
```

```
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
```

```
                "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

           "traits": {
                "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
              }
            },
            "SecretArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
               }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
              }
```

```
            },
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
         "type": "operation",
         "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/GetServiceSettings"

          },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
         "type": "structure",
         "members": {}
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
           "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
```

```
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
        },
        "UsageOperation": {
          "target": "smithy.api#String",

   "traits": {
            "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
          }
        },
        "ProductCode": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
          "traits": {
            "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
```

```
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
        "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
        },
        "SubscriptionProviderUpdateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
        },
        "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
        },
        "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {
```

```
    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager Linux Subscriptions",
          "arnNamespace": "license-manager-linux-subscriptions",
          "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
```

```
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
       "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
```

```
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }

    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "rules": [
                      {
                        "conditions": [
                          {
                            "fn": "booleanEquals",
                            "argv": [
                              {
                                "ref": "UseDualStack"
                              },
                              true
                            ]

}
                        ],
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                        "type": "error"
                      },
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": {
                            "ref": "Endpoint"
                          },
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"

                }
                ],
```

```
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

                                    "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
```

```
                    },
                    true
                ]
            }

    ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"

                },
                                            "supportsFIPS"
                                        ]
                                    }
                                ]
                            },
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],

        "rules": [
                                        {
```

```
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],

        "type": "tree"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
            "type": "error"
          }
        ],
        "type": "tree"
      },
      {

    "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {

      "fn": "booleanEquals",
                "argv": [
                  {
                    "fn": "getAttr",
                    "argv": [
```

```
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true

                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled
but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
```

```
                    },
                    true

                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {

                                    "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "rules": [

    {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"

                    }
                ],
                "type": "tree"
            },
```

```
                                    {
                                        "conditions": [],
                                        "error": "DualStack is enabled but this partition does not support DualStack",
                                        "type": "error"
                                    }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ],
        "type": "tree"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
```

```
    }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {

    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
          }

  },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
```

```
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {

  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
       },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
```

```
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
```

```
                {

  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }

                },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
```

```
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },

    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
```

```
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {

      "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",

          "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
        "version": "1.0"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
```

```
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
            "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "OrganizationIntegration": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
                "traits": {
                    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n            </li>\n
 <li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>InstanceType</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
 <code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
 <code>Status</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
 <code>UsageOperation</code>\n                </p>\n            </li>\n        </ul>\n        <p>For each filter, you can use
 one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n                <p>\n                    <code>contains</code>\n                </p>\n            </li>\n            <li>\n
 <p>\n                    <code>equals</code>\n                </p>\n            </li>\n            <li>\n
            <p>\n                    <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
                }
            },
```

```
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
          "type": "structure",
          "members": {
            "Instances": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
          },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n         <p>The valid names for this filter are:</p>\n         <ul>\n            <li>\n               <p>\n                  <code>Subscription</code>\n               </p>\n            </li>\n         </ul>\n         <p>The valid Operators for this filter are:</p>\n         <ul>\n            <li>\n               <p>\n                  <code>contains</code>\n               </p>\n            </li>\n            <li>\n               <p>\n                  <code>equals</code>\n               </p>\n            </li>\n            <li>\n               <p>\n
```

```
        <code>Notequal</code>\n              </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
```

```
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
          ],
          "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
             },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
```

```
      "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
               "smithy.api#required": {}
             }
           },
           "SecretArn": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've
stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat
Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
               "smithy.api#required": {}
             }
           },
           "Tags": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
             "traits": {

            "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
             }
           }
         },
         "traits": {
           "smithy.api#input": {}
         }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
         "type": "structure",
         "members": {
           "SubscriptionProviderSource": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
             }
           },
           "SubscriptionProviderArn": {
             "target": "smithy.api#String",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
             }
           },
           "SubscriptionProviderStatus": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
             "traits": {
               "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
```

```
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
                }
            },
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
                }
            },
```

```
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {

                "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "InProgress",
                  "value": "InProgress",
```

```
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }
        }
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
```

```
                {
                    "name": "RedHat",
                    "value": "RedHat",
                    "documentation": "RedHat subscription provider namespace"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
```

```
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
          "type": "map",
          "key":
{
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
          "type": "operation",
```

```
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input":
{}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
```

```
                "smithy.api#required":
{}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
            }
        },
        "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
        }

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }

  },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
```

```
            },
            "StatusMessage": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
               "traits": {
                  "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
               }
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
             }
          }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
             "message": {
                "target": "smithy.api#String"
             }
          },
          "traits": {
             "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
             "smithy.api#error": "client"
          }
       }
    }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.33 protobuf 1.36.10

1.33.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.34 aws-sdk-go-v2-internal-endpoints 2.7.11

1.34.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

```
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
```

```
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata
```

```
 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint

```
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
```

```
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
 spanInitializeStart
 InterceptExecution
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 spanInitializeEnd
 Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
```

```
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
```

```
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
```

```
 OperationInputValidation
 spanInitializeEnd
Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
```

```
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

       "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                     "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
 {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
               }
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
         "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
         "type": "structure",
         "members": {
            "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
          "smithy.api#error": "server"
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

      {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
```

```
            "uri": "/license-server/CreateLicenseServerEndpoint"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
```

```
            },
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
         "type": "union",
         "members": {
            "SecretsManagerCredentialsProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
               "traits": {
                  "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

   "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },

        "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
          "type": "structure",
          "members": {
            "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members":
```

```
{
            "Attribute": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
               }
            },
            "Operation": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
               }
            },
            "Value": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Value of the filter.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
         "type": "union",
         "members": {
            "ActiveDirectoryIdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
         }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
```

```
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                  "smithy.api#required": {}
               }
            },
            "Products": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                  "smithy.api#required": {}
               }
            },
            "LastStatusCheckDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date of the last status check.</p>"
```

```
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
           }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
          }
        },
```

```
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
```

```
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
```

```
 }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [

{
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },

              "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
```

```
    },
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
               "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },

true
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
```

```
                "rules": [
                  {
                    "conditions": [
                      {

        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },

                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsDualStack"
                            ]
                          }

                ]
                      }
                    ],
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                          "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
```

```
        ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [

{
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

            {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
```

```
                "conditions": [],
                "endpoint": {

          "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"

}
          ],
          "type": "tree"
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ],
          "rules": [
            {

    "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsDualStack"
```

```
                                    ]

                }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
```

```
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",

            "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
```

```
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                    }
                  },
                  "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                  },
                  "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                  "expect": {

 "endpoint": {
                      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
```

```
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
```

```
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
```

```
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

  "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

   "UseDualStack": false,
        "Endpoint": "https://example.com"
```

```
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
       }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
```

```
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
          },
          "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
              "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
          },
          "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
          },
          "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                "smithy.api#documentation": "<p>The <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code> resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
```

```
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
                    "value": "PROVISIONING_FAILED",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONED",
                    "value": "PROVISIONED",
                    "documentation": ""
                },
                {
```

```
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
```

```
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

     "members": {
            "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
     "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
          "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
         "type": "structure",
         "members": {
            "LicenseServerEndpoints": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits":
 {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
         },
         "errors": [
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
```

```
</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
```

```
you want to retrieve.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

  "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output":
 {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
```

```
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                "smithy.api#required": {}
              }
            },
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
```

```
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
 reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
 license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
```

```
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

          },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
          "type": "string",
```

```
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
      "type": "structure",
      "members": {
        "Endpoint": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
      "type": "union",

     "members": {
        "RdsSalSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }

    },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

         {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
```

```
 }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {

"min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
```

```
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

          }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
              "method": "POST",
              "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "IdentityProviderArn":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
        },
        "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                "smithy.api#required": {}
            }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
```

```
            "RemoveSubnets": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                  "smithy.api#required": {}
               }
            },
            "SecurityGroupId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
               "traits": {
                  "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
               }
            }
         },

         "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
```

```
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
```

```
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
```

```
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
```

```
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
```

```
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
```

```
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
```

```
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
```

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata
```

```
 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
```

```
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
```

```
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
```

```
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
```

```
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }
```

```
 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
```

```
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
```

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata

```
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if
err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
```

RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },

    {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                 "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
```

```
        "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
```

```
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"

        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
```

```
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
        "Region": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

  "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
```

```
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

     "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
```

```
        "traits": {
            "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
             },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
            }
        ],
        "traits": {
            "aws.api#service": {
                "sdkId": "License Manager Linux Subscriptions",
```

```
        "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id"
        ],
        "additionalExposedHeaders": [
          "x-amzn-errortype",
          "x-amzn-requestid",
          "x-amzn-trace-id"
        ]
      },
      "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
      "smithy.api#title": "AWS License Manager Linux Subscriptions",
      "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
          "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
          },
          "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
```

```
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }

],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
```

```
                    },
                    true
                ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
```

```
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {

                  "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }

],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"

},
                          "supportsFIPS"
                        ]
                      }
```

```
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [

                     {
                          "ref": "PartitionResult"
                      },
                      "supportsDualStack"
                    ]
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],

    "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],

            "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
```

```
                "type": "tree"
            },
            {

        "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

                "fn": "booleanEquals",
                            "argv": [
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true

                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                        "headers": {}
```

```
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS is enabled
but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true

        ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {

                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
```

```
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],
                "rules": [

        {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"

            }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "DualStack is enabled but this partition does not support DualStack",
              "type": "error"
            }
          ],
          "type": "tree"
        },
        {
          "conditions": [],
          "rules": [
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ],
          "type": "tree"
```

```
                        }
                      ],
                      "type": "tree"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "Invalid Configuration: Missing Region",
                  "type": "error"
                }
              ],
              "type": "tree"
            }
          ]
        },
        "smithy.rules#endpointTests": {
          "testCases": [
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

          }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
```

```
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                  }

        },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {

          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
```

```
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

      },
      "params": {
```

```
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
```

```
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },

        {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {

          "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
```

```
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "Missing region",

                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
```

```
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
```

```
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n
</p>\n            </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n            </li>\n
<li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>InstanceType</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>Status</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
<code>UsageOperation</code>\n                </p>\n            </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n                <p>\n                    <code>contains</code>\n                </p>\n            </li>\n            <li>\n
<p>\n                    <code>equals</code>\n                </p>\n            </li>\n            <li>\n
                <p>\n                    <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
```

```
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
```

```
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
```

```
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
```

```
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "EQUAL",
                    "value": "Equal",
                    "documentation": "Equal operator"
                },
                {
                    "name": "NOT_EQUAL",
                    "value": "NotEqual",
                    "documentation": "Not equal operator"
```

```
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
          "smithy.api#length": {
            "min": 1,
            "max": 20
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {

      "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
```

```
            },

           {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
         "type": "structure",
         "members": {
            "SubscriptionProviderSource": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                  "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                  "smithy.api#required": {}
               }
            },
            "SecretArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
               "traits": {

          "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
               }
            }
         },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
          }
        },
        "SubscriptionProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
 subscription provider \n\t\t\tthat you requested.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
 Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
```

```
            },
            "SecretArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
               }
            },
            "SubscriptionProviderStatus": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
               }
            },
            "SubscriptionProviderStatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
               }
            },
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {

                  "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
         "type": "structure",
```

```
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with
Amazon EC2, Bring Your Own\n        Subscription model (BYOS), or from the Amazon Web Services Marketplace.
Certain subscriptions may use licensing from the\n        Amazon Web Services Marketplace as well as OS licensing
from Amazon EC2 or BYOS.</p>"
                }
            },
            "InstanceCount": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
                "traits": {
                    "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

     "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RedHat",
                    "value": "RedHat",
                    "documentation": "RedHat subscription provider namespace"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
```

```
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
               "method": "PUT",
               "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
         "type": "structure",
         "members": {
            "resourceArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                   "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            },
            "tags": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
         "type": "structure",
         "members": {},
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
         "type": "map",
         "key":
{
            "target": "smithy.api#String"
         },
         "value": {
```

```
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
      "type": "structure",

  "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        },
        "tagKeys": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input":
{}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }

    },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
         }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.35 utmps 0.1.3.1-r0

1.35.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed
to make sure that you
have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work
which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim
copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code
or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy
the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.
Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to

address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version
number of this License which applies to it and "any
later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE
LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest

possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the
License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands
`show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.
Automatically generated by apkbuild-cpan, template 4
Contributor: Valery Kartel <valery.kartel@gmail.com>
Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
  INSTALLDIRS=vendor \
  NO_PACKLIST=1 \
  NO_PERLLOCAL=1
 make
}

check() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 make test
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="

```
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
  BSD-Resource-1.2911.tar.gz
"
# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
 0001-bsd-games-scores-dir.patch
 "
options="!check" # no tests

build() {
 # some GNU autoconf options are ignored, but it works
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr \
  --sysconfdir=/etc \
  --mandir=/usr/share/man \
  --scoredir=/usr/share/$pkgname
 make
}

package() {
 make DESTDIR="$pkgdir" install
 install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
  bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6  0001-bsd-games-scores-dir.patch
"
```

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION
 WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2
 "

builddir="$srcdir"/libart_lgpl-$pkgver

prepare() {
 update_config_sub
 default_prepare
}

build() {
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr
 make
}

package() {

```
 make DESTDIR="$pkgdir" install
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020  libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

Copyright  1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>
# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>
pkgname=spdx-licenses
pkgver=3.26.0
pkgrel=0
pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdfnt rdfturtle rdfxml template text"

for type in $_types; do
```

```
 subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
 mkdir -p "$pkgdir"
}

_subpkg() {
 local type=${subpkgname/$pkgname-/}
 pkgdesc="$pkgdesc ($type)"
 install_if="$pkgname=$pkgver-r$pkgrel"
 mkdir -p "$subpkgdir"/usr/share/spdx
 cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}

list() {
 pkgdesc="$pkgdesc (licence list)"
 mkdir -p "$subpkgdir"/usr/share/spdx
 local i; for i in "$builddir"/text/*.txt;
 do
  local license=${i##*/}
  echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
 done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a  license-list-data-3.26.0.tar.gz
"
# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
 cdefs.h
 queue.h
 tree.h
"
builddir="$srcdir"
options="!check" # just headers

package() {
 mkdir -p "$pkgdir"
```

```
 install -Dm644 -t "$pkgdir"/usr/include/sys \
  cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178  cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750  queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3  tree.h
"
```

Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)
The text rotation code in w_rottext.c was written by Alan Richardson.
 The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker
The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick
The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:
"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95
http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html

```
# Contributor: Dhruvin Gandhi <contact@dhru.vin>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=py3-license-expression
_pkgname=license_expression
pkgver=30.4.1
pkgrel=0
pkgdesc="Library to parse, compare, simplify and normalize license expressions"
url="https://github.com/nexB/license-expression"
```

```
arch="noarch"
license="Apache-2.0"
depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir/$_pkgname-$pkgver"

build() {
 gpep517 build-wheel \
  --wheel-dir .dist \
  --output-fd 3 3>&1 >&2
}

check() {
 python3 -m venv --clear --without-pip --system-site-packages .testenv
 .testenv/bin/python3 -m installer .dist/*.whl
 .testenv/bin/python3 -m pytest
}

package() {
 python3 -m installer -d "$pkgdir" \
  .dist/*.whl
}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
  license_expression-30.4.1.tar.gz
"
/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
* Copyright (c) 1991, 1993
* The Regents of the University of California.  All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
```

```
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
*/

#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_

/*
* This file defines five types of data structures: singly-linked lists,
* lists, simple queues, tail queues, and circular queues.
*
* A singly-linked list is headed by a single forward pointer. The
* elements are singly linked for minimum space and
 pointer manipulation
* overhead at the expense of O(n) removal for arbitrary elements. New
* elements can be added to the list after an existing element or at the
* head of the list.  Elements being removed from the head of the list
* should use the explicit macro for this purpose for optimum
* efficiency. A singly-linked list may only be traversed in the forward
* direction.  Singly-linked lists are ideal for applications with large
* datasets and few or no removals or for implementing a LIFO queue.
*
* A list is headed by a single forward pointer (or an array of forward
* pointers for a hash table header). The elements are doubly linked
* so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before
* or after an existing element or at the head of the list. A list
* may only be traversed in the forward direction.
*
* A simple queue is headed by a pair of pointers, one the head of the
* list and the
 other to the tail of the list. The elements are singly
* linked to save space, so elements can only be removed from the
* head of the list. New elements can be added to the list after
* an existing element, at the head of the list, or at the end of the
```

```
* list. A simple queue may only be traversed in the forward direction.
*
* A tail queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before or
* after an existing element, at the head of the list, or at the end of
* the list. A tail queue may be traversed in either direction.
*
* A circle queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements
 can be added to the list before or after
* an existing element, at the head of the list, or at the end of the list.
* A circle queue may be traversed in either direction, but has a more
* complex end of list detection.
*
* For details on the use of these macros, see the queue(3) manual page.
*/

/*
* Include the definition of NULL only on NetBSD because sys/null.h
* is not available elsewhere.  This conditional makes the header
* portable and it can simply be dropped verbatim into any system.
* The caveat is that on other systems some other header
* must provide NULL before the macros can be used.
*/
#ifdef __NetBSD__
#include <sys/null.h>
#endif

#if defined(QUEUEDEBUG)
# if defined(_KERNEL)
#  define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
#  include <err.h>
#  define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
* Singly-linked List definitions.
*/
#define SLIST_HEAD(name, type)      \
struct name {       \
 struct type *slh_first; /* first
```

```
 element */   \
}

#define SLIST_HEAD_INITIALIZER(head)     \
 { NULL }

#define SLIST_ENTRY(type)      \
struct {        \
 struct type *sle_next; /* next element */   \
}

/*
* Singly-linked List access methods.
*/
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head)  NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field)     \
 for((var) = (head)->slh_first;     \
     (var) != SLIST_END(head);     \
     (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = SLIST_FIRST((head));    \
     (var) != SLIST_END(head) &&     \
     ((tvar) = SLIST_NEXT((var), field), 1);   \
     (var) = (tvar))

/*
* Singly-linked List functions.
*/
#define SLIST_INIT(head) do {      \
 (head)->slh_first = SLIST_END(head);    \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do {   \
 (elm)->field.sle_next = (slistelm)->field.sle_next;  \
 (slistelm)->field.sle_next
 = (elm);    \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do {   \
 (elm)->field.sle_next = (head)->slh_first;   \
 (head)->slh_first = (elm);     \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_REMOVE_AFTER(slistelm, field) do {   \
 (slistelm)->field.sle_next =     \
    SLIST_NEXT(SLIST_NEXT((slistelm), field), field);  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do {    \
 (head)->slh_first = (head)->slh_first->field.sle_next;  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do {   \
 if ((head)->slh_first == (elm)) {    \
  SLIST_REMOVE_HEAD((head), field);   \
 }       \
 else {       \
  struct type *curelm = (head)->slh_first;  \
  while(curelm->field.sle_next != (elm))   \
   curelm = curelm->field.sle_next;  \
  curelm->field.sle_next =    \
     curelm->field.sle_next->field.sle_next;  \
 }       \
} while (/*CONSTCOND*/0)


/*
* List definitions.
 */
#define LIST_HEAD(name, type)      \
struct name {       \
 struct type *lh_first; /* first element */   \
}

#define LIST_HEAD_INITIALIZER(head)     \
 { NULL }

#define LIST_ENTRY(type)      \
struct {       \
 struct type *le_next; /* next element */   \
 struct type **le_prev; /* address of previous next element */ \
}

/*
* List access methods.
*/
#define LIST_FIRST(head)  ((head)->lh_first)
#define LIST_END(head)   NULL
#define LIST_EMPTY(head)  ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field)  ((elm)->field.le_next)
```

```
#define LIST_FOREACH(var, head, field)     \
 for ((var) = ((head)->lh_first);    \
     (var) != LIST_END(head);     \
     (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = LIST_FIRST((head));    \
     (var) != LIST_END(head) &&     \
     ((tvar) = LIST_NEXT((var), field), 1);   \
     (var) = (tvar))

#define LIST_MOVE(head1, head2) do {     \
 LIST_INIT((head2));      \
 if (!LIST_EMPTY((head1)))
 {     \
  (head2)->lh_first = (head1)->lh_first;   \
  LIST_INIT((head1));     \
 }       \
} while (/*CONSTCOND*/0)

/*
* List functions.
*/
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)   \
 if ((head)->lh_first &&      \
     (head)->lh_first->field.le_prev != &(head)->lh_first) \
  QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field)     \
 if ((elm)->field.le_next &&     \
     (elm)->field.le_next->field.le_prev !=   \
     &(elm)->field.le_next)     \
  QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm),  \
      __FILE__, __LINE__);    \
 if (*(elm)->field.le_prev != (elm))    \
  QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm),  \
      __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)    \
 (elm)->field.le_next = (void *)1L;    \
 (elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,
 field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif
```

```
#define LIST_INIT(head) do {      \
 (head)->lh_first = LIST_END(head);    \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do {   \
 QUEUEDEBUG_LIST_OP((listelm), field)    \
 if (((elm)->field.le_next = (listelm)->field.le_next) !=  \
     LIST_END(head))      \
  (listelm)->field.le_next->field.le_prev =  \
      &(elm)->field.le_next;    \
 (listelm)->field.le_next = (elm);    \
 (elm)->field.le_prev = &(listelm)->field.le_next;  \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do {   \
 QUEUEDEBUG_LIST_OP((listelm), field)    \
 (elm)->field.le_prev = (listelm)->field.le_prev;  \
 (elm)->field.le_next = (listelm);    \
 *(listelm)->field.le_prev = (elm);    \
 (listelm)->field.le_prev = &(elm)->field.le_next;  \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do {    \
 QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field)  \
 if (((elm)->field.le_next
 = (head)->lh_first) != LIST_END(head))\
  (head)->lh_first->field.le_prev = &(elm)->field.le_next;\
 (head)->lh_first = (elm);     \
 (elm)->field.le_prev = &(head)->lh_first;   \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do {     \
 QUEUEDEBUG_LIST_OP((elm), field)    \
 if ((elm)->field.le_next != NULL)    \
  (elm)->field.le_next->field.le_prev =    \
      (elm)->field.le_prev;    \
 *(elm)->field.le_prev = (elm)->field.le_next;   \
 QUEUEDEBUG_LIST_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do {    \
 if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
  (elm2)->field.le_next->field.le_prev =   \
      &(elm2)->field.le_next;    \
 (elm2)->field.le_prev = (elm)->field.le_prev;   \
 *(elm2)->field.le_prev = (elm2);    \
 QUEUEDEBUG_LIST_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)
```

```
/*
* Simple queue definitions.
*/
#define SIMPLEQ_HEAD(name, type)     \
struct name {        \
 struct
 type *sqh_first; /* first element */   \
 struct type **sqh_last; /* addr of last next element */  \
}

#define SIMPLEQ_HEAD_INITIALIZER(head)     \
 { NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type)      \
struct {        \
 struct type *sqe_next; /* next element */   \
}

/*
* Simple queue access methods.
*/
#define SIMPLEQ_FIRST(head)  ((head)->sqh_first)
#define SIMPLEQ_END(head)  NULL
#define SIMPLEQ_EMPTY(head)  ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->sqh_first);    \
     (var) != SIMPLEQ_END(head);     \
     (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next)   \
 for ((var) = ((head)->sqh_first);    \
     (var) != SIMPLEQ_END(head) &&    \
     ((next = ((var)->field.sqe_next)), 1);   \
     (var) = (next))

/*
* Simple queue functions.
*/
#define SIMPLEQ_INIT(head) do {      \
 (head)->sqh_first = NULL;     \
 (head)->sqh_last
 = &(head)->sqh_first;    \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do {   \
```

```
 if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
  (head)->sqh_last = &(elm)->field.sqe_next;  \
 (head)->sqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do {   \
 (elm)->field.sqe_next = NULL;     \
 *(head)->sqh_last = (elm);     \
 (head)->sqh_last = &(elm)->field.sqe_next;   \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL)\
  (head)->sqh_last = &(elm)->field.sqe_next;  \
 (listelm)->field.sqe_next = (elm);    \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do {    \
 if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
  (head)->sqh_last = &(head)->sqh_first;   \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do {   \
 if
 (((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
    == NULL)       \
  (head)->sqh_last = &(elm)->field.sqe_next;  \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do {   \
 if ((head)->sqh_first == (elm)) {    \
  SIMPLEQ_REMOVE_HEAD((head), field);   \
 } else {       \
  struct type *curelm = (head)->sqh_first;  \
  while (curelm->field.sqe_next != (elm))   \
   curelm = curelm->field.sqe_next;  \
  if ((curelm->field.sqe_next =    \
   curelm->field.sqe_next->field.sqe_next) == NULL) \
      (head)->sqh_last = &(curelm)->field.sqe_next; \
 }       \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do {    \
 if (!SIMPLEQ_EMPTY((head2))) {     \
  *(head1)->sqh_last = (head2)->sqh_first;  \
  (head1)->sqh_last = (head2)->sqh_last;  \
  SIMPLEQ_INIT((head2));     \
 }       \
```

```
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field)     \
 (SIMPLEQ_EMPTY((head)) ?      \
  NULL :       \
        ((struct type
 *)(void *)    \
  ((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
* Tail queue definitions.
*/
#define _TAILQ_HEAD(name, type, qual)     \
struct name {        \
 qual type *tqh_first;  /* first element */  \
 qual type *qual *tqh_last; /* addr of last next element */ \
}
#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head)     \
 { TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual)     \
struct {        \
 qual type *tqe_next;  /* next element */  \
 qual type *qual *tqe_prev; /* address of previous next element */\
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
* Tail queue access methods.
*/
#define TAILQ_FIRST(head)  ((head)->tqh_first)
#define TAILQ_END(head)   (NULL)
#define TAILQ_NEXT(elm, field)  ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
 (*(((struct headname *)(void *)((head)->tqh_last))->tqh_last))
#define TAILQ_PREV(elm, headname, field) \
 (*(((struct headname
 *)(void *)((elm)->field.tqe_prev))->tqh_last))
#define TAILQ_EMPTY(head)  (TAILQ_FIRST(head) == TAILQ_END(head))


#define TAILQ_FOREACH(var, head, field)     \
 for ((var) = ((head)->tqh_first);    \
     (var) != TAILQ_END(head);     \
     (var) = ((var)->field.tqe_next))
```

```
#define TAILQ_FOREACH_SAFE(var, head, field, next)   \
 for ((var) = ((head)->tqh_first);    \
     (var) != TAILQ_END(head) &&     \
     ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field)  \
 for ((var) = TAILQ_LAST((head), headname);   \
     (var) != TAILQ_END(head);     \
     (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
 for ((var) = TAILQ_LAST((head), headname);   \
     (var) != TAILQ_END(head) &&     \
     ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
* Tail queue functions.
*/
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head,
 elm, field)   \
 if ((head)->tqh_first &&     \
     (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
  QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)   \
 if (*(head)->tqh_last != NULL)     \
  QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field)     \
 if ((elm)->field.tqe_next &&     \
     (elm)->field.tqe_next->field.tqe_prev !=   \
     &(elm)->field.tqe_next)     \
  QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
      __FILE__, __LINE__);    \
 if (*(elm)->field.tqe_prev != (elm))    \
  QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)   \
 if ((elm)->field.tqe_next == NULL &&    \
     (head)->tqh_last != &(elm)->field.tqe_next)   \
  QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d",\
      (head), (elm), __FILE__,
 __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)    \
 (elm)->field.tqe_next = (void *)1L;    \
 (elm)->field.tqe_prev = (void *)1L;
#else
```

```
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do {      \
 (head)->tqh_first = TAILQ_END(head);    \
 (head)->tqh_last = &(head)->tqh_first;    \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do {    \
 QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field)  \
 if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head))\
  (head)->tqh_first->field.tqe_prev =    \
     &(elm)->field.tqe_next;    \
 else       \
  (head)->tqh_last = &(elm)->field.tqe_next;  \
 (head)->tqh_first = (elm);     \
 (elm)->field.tqe_prev = &(head)->tqh_first;   \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head,
 elm, field) do {    \
 QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field)  \
 (elm)->field.tqe_next = TAILQ_END(head);    \
 (elm)->field.tqe_prev = (head)->tqh_last;   \
 *(head)->tqh_last = (elm);     \
 (head)->tqh_last = &(elm)->field.tqe_next;   \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 QUEUEDEBUG_TAILQ_OP((listelm), field)    \
 if (((elm)->field.tqe_next = (listelm)->field.tqe_next) !=  \
    TAILQ_END(head))      \
  (elm)->field.tqe_next->field.tqe_prev =    \
     &(elm)->field.tqe_next;    \
 else       \
  (head)->tqh_last = &(elm)->field.tqe_next;  \
 (listelm)->field.tqe_next = (elm);    \
 (elm)->field.tqe_prev = &(listelm)->field.tqe_next;  \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do {    \
 QUEUEDEBUG_TAILQ_OP((listelm), field)    \
 (elm)->field.tqe_prev = (listelm)->field.tqe_prev;  \
 (elm)->field.tqe_next = (listelm);    \
```

```
 *(listelm)->field.tqe_prev = (elm);    \
 (listelm)->field.tqe_prev
 = &(elm)->field.tqe_next;  \
} while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do {    \
 QUEUEDEBUG_TAILQ_PREREMOVE((head), (elm), field)  \
 QUEUEDEBUG_TAILQ_OP((elm), field)    \
 if (((elm)->field.tqe_next) != TAILQ_END(head))   \
  (elm)->field.tqe_next->field.tqe_prev =   \
     (elm)->field.tqe_prev;    \
 else       \
  (head)->tqh_last = (elm)->field.tqe_prev;  \
 *(elm)->field.tqe_prev = (elm)->field.tqe_next;   \
 QUEUEDEBUG_TAILQ_POSTREMOVE((elm), field);   \
} while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do {   \
      if (((elm2)->field.tqe_next = (elm)->field.tqe_next) !=  \
    TAILQ_END(head))        \
            (elm2)->field.tqe_next->field.tqe_prev =  \
               &(elm2)->field.tqe_next;    \
      else       \
            (head)->tqh_last = &(elm2)->field.tqe_next;  \
      (elm2)->field.tqe_prev = (elm)->field.tqe_prev;   \
      *(elm2)->field.tqe_prev = (elm2);    \
 QUEUEDEBUG_TAILQ_POSTREMOVE((elm),
 field);   \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do {    \
 if (!TAILQ_EMPTY(head2)) {     \
  *(head1)->tqh_last = (head2)->tqh_first;  \
  (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
  (head1)->tqh_last = (head2)->tqh_last;   \
  TAILQ_INIT((head2));     \
 }       \
} while (/*CONSTCOND*/0)

/*
* Singly-linked Tail queue declarations.
*/
#define STAILQ_HEAD(name, type)      \
struct name {       \
 struct type *stqh_first; /* first element */  \
 struct type **stqh_last; /* addr of last next element */ \
 }
```

```
#define STAILQ_HEAD_INITIALIZER(head)     \
 { NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type)      \
struct {        \
 struct type *stqe_next; /* next element */   \
}

/*
* Singly-linked Tail queue access methods.
*/
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head)
 == STAILQ_END(head))

/*
* Singly-linked Tail queue functions.
*/
#define STAILQ_INIT(head) do {      \
 (head)->stqh_first = NULL;     \
 (head)->stqh_last = &(head)->stqh_first;    \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do {   \
 if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
  (head)->stqh_last = &(elm)->field.stqe_next;  \
 (head)->stqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do {   \
 (elm)->field.stqe_next = NULL;     \
 *(head)->stqh_last = (elm);     \
 (head)->stqh_last = &(elm)->field.stqe_next;   \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL)\
  (head)->stqh_last = &(elm)->field.stqe_next;  \
 (listelm)->field.stqe_next = (elm);    \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE_HEAD(head, field) do {    \
 if (((head)->stqh_first = (head)->stqh_first->field.stqe_next)
 == NULL) \
  (head)->stqh_last = &(head)->stqh_first;   \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_REMOVE(head, elm, type, field) do {   \
 if ((head)->stqh_first == (elm)) {    \
  STAILQ_REMOVE_HEAD((head), field);   \
 } else {       \
  struct type *curelm = (head)->stqh_first;  \
  while (curelm->field.stqe_next != (elm))   \
   curelm = curelm->field.stqe_next;  \
  if ((curelm->field.stqe_next =    \
   curelm->field.stqe_next->field.stqe_next) == NULL) \
      (head)->stqh_last = &(curelm)->field.stqe_next; \
 }       \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->stqh_first);    \
  (var);       \
  (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = STAILQ_FIRST((head));    \
    (var) && ((tvar) = STAILQ_NEXT((var), field), 1);  \
    (var) = (tvar))

#define STAILQ_CONCAT(head1, head2) do {    \
 if (!STAILQ_EMPTY((head2))) {     \
  *(head1)->stqh_last
 = (head2)->stqh_first;  \
  (head1)->stqh_last = (head2)->stqh_last;  \
  STAILQ_INIT((head2));     \
 }       \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field)     \
 (STAILQ_EMPTY((head)) ?      \
  NULL :       \
       ((struct type *)(void *)    \
  ((char *)((head)->stqh_last) - offsetof(struct type, field))))


#ifndef _KERNEL
/*
* Circular queue definitions. Do not use. We still keep the macros
* for compatibility but because of pointer aliasing issues their use
* is discouraged!
*/

/*
```

```
* __launder_type():  We use this ugly hack to work around the the compiler
* noticing that two types may not alias each other and elide tests in code.
* We hit this in the CIRCLEQ macros when comparing 'struct name *' and
* 'struct type *' (see CIRCLEQ_HEAD()).  Modern compilers (such as GCC
* 4.8) declare these comparisons as always false, causing the code to
* not run as designed.
*
* This hack is only to be used for comparisons and thus can be fully const.
 * Do not use for assignment.
*
* If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
* this by changing the head/tail sentinal values, but see the note above
* this one.
*/
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
 __asm __volatile("" : "+r" (__x));
 return __x;
}

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)    \
 if ((head)->cqh_first != CIRCLEQ_ENDC(head) &&   \
    (head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
  QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
       __FILE__, __LINE__);    \
 if ((head)->cqh_last != CIRCLEQ_ENDC(head) &&   \
    (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
  QUEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
       __FILE__, __LINE__);
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)   \
 if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) {  \
  if ((head)->cqh_last != (elm))    \
   QUEUEDEBUG_ABORT("CIRCLEQ
 elm last %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 } else {       \
  if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
   QUEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 }       \
 if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) {  \
  if ((head)->cqh_first != (elm))    \
   QUEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 } else {       \
```

```
 if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
  QUEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 }
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)   \
 (elm)->field.cqe_next = (void *)1L;    \
 (elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type)     \
struct name {        \
 struct type *cqh_first;  /*
 first element */  \
 struct type *cqh_last;  /* last element */  \
}

#define CIRCLEQ_HEAD_INITIALIZER(head)     \
 { CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type)      \
struct {        \
 struct type *cqe_next;  /* next element */  \
 struct type *cqe_prev;  /* previous element */  \
}

/*
* Circular queue functions.
*/
#define CIRCLEQ_INIT(head) do {      \
 (head)->cqh_first = CIRCLEQ_END(head);    \
 (head)->cqh_last = CIRCLEQ_END(head);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field)  \
 (elm)->field.cqe_next = (listelm)->field.cqe_next;  \
 (elm)->field.cqe_prev = (listelm);    \
 if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head))  \
  (head)->cqh_last = (elm);    \
 else        \
  (listelm)->field.cqe_next->field.cqe_prev = (elm); \
 (listelm)->field.cqe_next = (elm);    \
} while (/*CONSTCOND*/0)
```

```
#define CIRCLEQ_INSERT_BEFORE(head,
 listelm, elm, field) do {  \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field)  \
 (elm)->field.cqe_next = (listelm);    \
 (elm)->field.cqe_prev = (listelm)->field.cqe_prev;  \
 if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head))  \
  (head)->cqh_first = (elm);    \
 else       \
  (listelm)->field.cqe_prev->field.cqe_next = (elm); \
 (listelm)->field.cqe_prev = (elm);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do {   \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 (elm)->field.cqe_next = (head)->cqh_first;   \
 (elm)->field.cqe_prev = CIRCLEQ_END(head);   \
 if ((head)->cqh_last == CIRCLEQ_ENDC(head))   \
  (head)->cqh_last = (elm);    \
 else       \
  (head)->cqh_first->field.cqe_prev = (elm);  \
 (head)->cqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do {   \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 (elm)->field.cqe_next
 = CIRCLEQ_END(head);   \
 (elm)->field.cqe_prev = (head)->cqh_last;   \
 if ((head)->cqh_first == CIRCLEQ_ENDC(head))   \
  (head)->cqh_first = (elm);    \
 else       \
  (head)->cqh_last->field.cqe_next = (elm);  \
 (head)->cqh_last = (elm);     \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_REMOVE(head, elm, field) do {    \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field)   \
 if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head))  \
  (head)->cqh_last = (elm)->field.cqe_prev;  \
 else       \
  (elm)->field.cqe_next->field.cqe_prev =   \
     (elm)->field.cqe_prev;    \
 if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head))  \
  (head)->cqh_first = (elm)->field.cqe_next;  \
 else       \
```

```
 (elm)->field.cqe_prev->field.cqe_next =   \
     (elm)->field.cqe_next;    \
 QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->cqh_first);    \
  (var) != CIRCLEQ_ENDC(head);    \
  (var)
 = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field)   \
 for ((var) = ((head)->cqh_last);    \
  (var) != CIRCLEQ_ENDC(head);    \
  (var) = ((var)->field.cqe_prev))

/*
* Circular queue access methods.
*/
#define CIRCLEQ_FIRST(head)  ((head)->cqh_first)
#define CIRCLEQ_LAST(head)  ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head)  (__launder_type(head))
/* For assignments */
#define CIRCLEQ_END(head)  ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head)      \
   (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field)    \
 (((elm)->field.cqe_next == CIRCLEQ_ENDC(head))   \
     ? ((head)->cqh_first)     \
     : (elm->field.cqe_next))
#define CIRCLEQ_LOOP_PREV(head, elm, field)    \
 (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head))   \
     ? ((head)->cqh_last)     \
     :
 (elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
```

```
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright"                  # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
 gpep517 build-wheel \
  --wheel-dir .dist \
  --output-fd 3 3>&1 >&2
}

package() {
 gpep517 install-wheel --destdir "$pkgdir" \
  .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
  flake8-copyright-0.2.4.tar.gz
"
#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef  __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
```

```
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__ ((__nothrow__))
# define __NTH(fct) __attribute__ ((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct)     fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
 default_prepare

 cargo fetch --target="$CTARGET" --locked
}

build() {
 cargo auditable build --release --frozen
}

check() {
 cargo test --frozen
}

package() {
 install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"
/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
```

```
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
 AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y)      \
 ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) ||   \
  (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

/*
* This file defines data structures for different types of trees:
* splay trees and red-black trees.
*
```

```
* A splay tree is a self-organizing
 data structure.  Every operation
* on the tree causes a splay to happen.  The splay moves the requested
* node to the root of the tree and partly rebalances it.
*
* This has the benefit that request locality causes faster lookups as
* the requested nodes move to the top of the tree.  On the other hand,
* every lookup causes memory writes.
*
* The Balance Theorem bounds the total access time for m operations
* and n inserts on an initially empty tree as O((m + n)lg n).  The
* amortized cost for a sequence of m accesses to a splay tree is O(lg n);
*
* A red-black tree is a binary search tree with the node color as an
* extra attribute.  It fulfills a set of conditions:
* - every search path from the root to a leaf consists of the
*   same number of black nodes,
* - each red node (except for the root) has a black parent,
* - each leaf node is black.
*
* Every operation on a red-black tree is bounded as O(lg n).
* The maximum height of a red-black tree is 2lg (n+1).
 */

#define SPLAY_HEAD(name, type)      \
struct name {       \
 struct type *sph_root; /* root of the tree */   \
}

#define SPLAY_INITIALIZER(root)     \
 { NULL }

#define SPLAY_INIT(root) do {      \
 (root)->sph_root = NULL;     \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type)      \
struct {       \
 struct type *spe_left; /* left element */   \
 struct type *spe_right; /* right element */   \
}

#define SPLAY_LEFT(elm, field)  (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field)  (elm)->field.spe_right
#define SPLAY_ROOT(head)  (head)->sph_root
#define SPLAY_EMPTY(head)  (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
```

```
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do {   \
 SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
 SPLAY_RIGHT(tmp, field) = (head)->sph_root;   \
 (head)->sph_root = tmp;      \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do {   \
 SPLAY_RIGHT((head)->sph_root,
 field) = SPLAY_LEFT(tmp, field); \
 SPLAY_LEFT(tmp, field) = (head)->sph_root;   \
 (head)->sph_root = tmp;      \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do {    \
 SPLAY_LEFT(tmp, field) = (head)->sph_root;   \
 tmp = (head)->sph_root;      \
 (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do {    \
 SPLAY_RIGHT(tmp, field) = (head)->sph_root;   \
 tmp = (head)->sph_root;      \
 (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do {  \
 SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
 SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field);\
 SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
 SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
 SPLAY_PROTOTYPE(name, type, field, cmp)    \
void name##_SPLAY(struct name *, struct type *);   \
void name##_SPLAY_MINMAX(struct name *, int);    \
struct type *name##_SPLAY_INSERT(struct name *, struct type *);  \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *);  \
        \
/* Finds the node with the same key as elm */    \
static __inline struct type *      \
name##_SPLAY_FIND(struct name *head, struct type *elm)   \
{         \
 if (SPLAY_EMPTY(head))      \
  return(NULL);      \
 name##_SPLAY(head, elm);     \
```

```
 if ((cmp)(elm, (head)->sph_root) == 0)    \
  return (head->sph_root);    \
 return (NULL);       \
}         \
        \
static __inline _sys_tree_h_unused struct type *   \
name##_SPLAY_NEXT(struct name *head, struct type *elm)   \
{         \
 name##_SPLAY(head, elm);     \
 if (SPLAY_RIGHT(elm, field) != NULL) {    \
  elm = SPLAY_RIGHT(elm, field);    \
  while (SPLAY_LEFT(elm, field) != NULL) {  \
   elm = SPLAY_LEFT(elm, field);   \
  }       \
 }
 else        \
  elm = NULL;      \
 return (elm);       \
}         \
        \
static _sys_tree_h_unused __inline struct type *   \
name##_SPLAY_MIN_MAX(struct name *head, int val)   \
{         \
 name##_SPLAY_MINMAX(head, val);     \
       return (SPLAY_ROOT(head));     \
}

/* Main splay operation.
* Moves node close to the key of elm to top
*/
#define SPLAY_GENERATE(name, type, field, cmp)    \
struct type *        \
name##_SPLAY_INSERT(struct name *head, struct type *elm)  \
{         \
   if (SPLAY_EMPTY(head)) {      \
     SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
   } else {        \
     int __comp;       \
     name##_SPLAY(head, elm);     \
     __comp = (cmp)(elm, (head)->sph_root);   \
     if(__comp < 0) {      \
      SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field);\
      SPLAY_RIGHT(elm, field) = (head)->sph_root;  \
      SPLAY_LEFT((head)->sph_root, field) = NULL;  \
     } else if (__comp > 0) {     \
      SPLAY_RIGHT(elm,
 field) = SPLAY_RIGHT((head)->sph_root, field);\
      SPLAY_LEFT(elm, field) = (head)->sph_root;  \
```

```
      SPLAY_RIGHT((head)->sph_root, field) = NULL; \
     } else      \
      return ((head)->sph_root);    \
   }        \
   (head)->sph_root = (elm);      \
   return (NULL);       \
}        \
        \
struct type *        \
name##_SPLAY_REMOVE(struct name *head, struct type *elm)  \
{        \
 struct type *__tmp;      \
 if (SPLAY_EMPTY(head))      \
  return (NULL);      \
 name##_SPLAY(head, elm);     \
 if ((cmp)(elm, (head)->sph_root) == 0) {   \
  if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
   (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
  } else {      \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
   name##_SPLAY(head, elm);   \
   SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
  }       \
 return (elm);      \
 }        \
 return (NULL);       \
}        \
        \
void         \
name##_SPLAY(struct
 name *head, struct type *elm)   \
{        \
 struct type __node, *__left, *__right, *__tmp;   \
 int __comp;       \
\
 SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
 __left = __right = &__node;     \
\
 while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) {  \
  if (__comp < 0) {     \
   __tmp = SPLAY_LEFT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if ((cmp)(elm, __tmp) < 0){   \
    SPLAY_ROTATE_RIGHT(head, __tmp, field); \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
     break;    \
```

```
  }      \
  SPLAY_LINKLEFT(head, __right, field);  \
 } else if (__comp > 0) {    \
  __tmp = SPLAY_RIGHT((head)->sph_root, field); \
  if (__tmp == NULL)    \
   break;     \
  if ((cmp)(elm, __tmp) > 0){    \
   SPLAY_ROTATE_LEFT(head, __tmp, field); \
   if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
    break;    \
  }      \
  SPLAY_LINKRIGHT(head, __left, field);  \
 }       \
}       \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field);  \
}        \
       \
/* Splay with either the minimum or the maximum element   \
* Used to find minimum or maximum element in tree.   \
*/        \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{        \
 struct type __node, *__left, *__right, *__tmp;   \
\
 SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
 __left = __right = &__node;     \
\
 while (1) {       \
  if (__comp < 0) {     \
   __tmp = SPLAY_LEFT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if (__comp < 0){    \
    SPLAY_ROTATE_RIGHT(head, __tmp, field); \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
     break;    \
   }      \
   SPLAY_LINKLEFT(head, __right, field);  \
  } else if (__comp > 0) {    \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if (__comp > 0) {    \
    SPLAY_ROTATE_LEFT(head, __tmp, field); \
    if (SPLAY_RIGHT((head)->sph_root,
 field) == NULL)\
     break;    \
```

```
  }     \
  SPLAY_LINKRIGHT(head, __left, field);  \
 }      \
}       \
 SPLAY_ASSEMBLE(head, &__node, __left, __right, field);  \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y)  name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y)  name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x)  (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x)  (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head)    \
 for ((x) = SPLAY_MIN(name, head);   \
     (x) != NULL;     \
     (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type)     \
struct name {       \
 struct type *rbh_root; /* root of the tree */   \
}

#define RB_INITIALIZER(root)     \
 {
 NULL }

#define RB_INIT(root) do {      \
 (root)->rbh_root = NULL;    \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED  1
#define RB_ENTRY(type)      \
struct {        \
 struct type *rbe_left;  /* left element */  \
 struct type *rbe_right;  /* right element */  \
 struct type *rbe_parent; /* parent element */  \
 int rbe_color;   /* node color */  \
}
```

```
#define RB_LEFT(elm, field)  (elm)->field.rbe_left
#define RB_RIGHT(elm, field)  (elm)->field.rbe_right
#define RB_PARENT(elm, field)  (elm)->field.rbe_parent
#define RB_COLOR(elm, field)  (elm)->field.rbe_color
#define RB_ROOT(head)   (head)->rbh_root
#define RB_EMPTY(head)   (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do {     \
 RB_PARENT(elm, field) = parent;     \
 RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL;  \
 RB_COLOR(elm, field) = RB_RED;     \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do {    \
 RB_COLOR(black, field) = RB_BLACK;    \
 RB_COLOR(red, field)
 = RB_RED;     \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do {   \
 (tmp) = RB_RIGHT(elm, field);     \
 if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
  RB_PARENT(RB_LEFT(tmp, field), field) = (elm);  \
 }       \
 RB_AUGMENT(elm);      \
 if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
  if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
   RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
  else      \
   RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
 } else       \
  (head)->rbh_root = (tmp);    \
 RB_LEFT(tmp, field) = (elm);     \
 RB_PARENT(elm, field) = (tmp);     \
 RB_AUGMENT(tmp);      \
 if ((RB_PARENT(tmp, field)))     \
  RB_AUGMENT(RB_PARENT(tmp, field));   \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do {   \
 (tmp) = RB_LEFT(elm, field);     \
 if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field))
 != NULL) { \
  RB_PARENT(RB_RIGHT(tmp, field), field) = (elm);  \
```

```
 }       \
 RB_AUGMENT(elm);      \
 if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
  if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
   RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
  else      \
   RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
 } else       \
  (head)->rbh_root = (tmp);    \
 RB_RIGHT(tmp, field) = (elm);     \
 RB_PARENT(elm, field) = (tmp);     \
 RB_AUGMENT(tmp);      \
 if ((RB_PARENT(tmp, field)))     \
  RB_AUGMENT(RB_PARENT(tmp, field));   \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp)    \
 RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp)   \
 RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr)  \
attr void name##_RB_INSERT_COLOR(struct name *, struct
 type *);  \
attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
attr struct type *name##_RB_INSERT(struct name *, struct type *); \
attr struct type *name##_RB_FIND(struct name *, struct type *);  \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *);   \
attr struct type *name##_RB_PREV(struct type *);   \
attr struct type *name##_RB_MINMAX(struct name *, int);   \
        \

/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp)    \
 RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp)   \
 RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr)  \
attr void       \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm)  \
{       \
 struct type
 *parent, *gparent, *tmp;    \
 while ((parent = RB_PARENT(elm, field)) != NULL &&  \
```

```
    RB_COLOR(parent, field) == RB_RED) {   \
  gparent = RB_PARENT(parent, field);   \
  if (parent == RB_LEFT(gparent, field)) {  \
   tmp = RB_RIGHT(gparent, field);   \
   if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent;    \
    continue;    \
   }      \
   if (RB_RIGHT(parent, field) == elm) {  \
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    tmp = parent;    \
    parent = elm;    \
    elm = tmp;    \
   }      \
   RB_SET_BLACKRED(parent, gparent, field); \
   RB_ROTATE_RIGHT(head, gparent, tmp, field); \
  } else {      \
   tmp = RB_LEFT(gparent, field);   \
   if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent;    \
    continue;    \
   }      \
   if (RB_LEFT(parent, field) == elm)
  {  \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent;    \
    parent = elm;    \
    elm = tmp;    \
   }      \
   RB_SET_BLACKRED(parent, gparent, field); \
   RB_ROTATE_LEFT(head, gparent, tmp, field); \
  }       \
 }       \
 RB_COLOR(head->rbh_root, field) = RB_BLACK;   \
}        \
        \
attr void        \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{        \
 struct type *tmp;      \
 while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
     elm != RB_ROOT(head)) {     \
  if (RB_LEFT(parent, field) == elm) {   \
   tmp = RB_RIGHT(parent, field);   \
```

```
  if (RB_COLOR(tmp, field) == RB_RED) {  \
   RB_SET_BLACKRED(tmp, parent, field); \
   RB_ROTATE_LEFT(head, parent, tmp, field);\
   tmp = RB_RIGHT(parent, field);  \
  }      \
  if ((RB_LEFT(tmp, field) == NULL ||  \
      RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
      (RB_RIGHT(tmp, field) == NULL ||  \
      RB_COLOR(RB_RIGHT(tmp,
field), field) == RB_BLACK)) {\
   RB_COLOR(tmp, field) = RB_RED;  \
   elm = parent;    \
   parent = RB_PARENT(elm, field);  \
  } else {     \
   if (RB_RIGHT(tmp, field) == NULL || \
       RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
    struct type *oleft;  \
    if ((oleft = RB_LEFT(tmp, field)) \
        != NULL)   \
     RB_COLOR(oleft, field) = RB_BLACK;\
    RB_COLOR(tmp, field) = RB_RED; \
    RB_ROTATE_RIGHT(head, tmp, oleft, field);\
    tmp = RB_RIGHT(parent, field); \
   }     \
   RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
   RB_COLOR(parent, field) = RB_BLACK; \
   if (RB_RIGHT(tmp, field))  \
    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
   RB_ROTATE_LEFT(head, parent, tmp, field);\
   elm = RB_ROOT(head);   \
   break;     \
  }      \
 } else {      \
  tmp = RB_LEFT(parent, field);   \
  if (RB_COLOR(tmp, field) == RB_RED) {  \
   RB_SET_BLACKRED(tmp, parent, field); \
   RB_ROTATE_RIGHT(head, parent, tmp, field);\
   tmp
= RB_LEFT(parent, field);  \
  }      \
  if ((RB_LEFT(tmp, field) == NULL ||  \
      RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
      (RB_RIGHT(tmp, field) == NULL ||  \
      RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
   RB_COLOR(tmp, field) = RB_RED;  \
   elm = parent;    \
   parent = RB_PARENT(elm, field);  \
  } else {     \
```

```
   if (RB_LEFT(tmp, field) == NULL || \
       RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
     struct type *oright;  \
     if ((oright = RB_RIGHT(tmp, field)) \
         != NULL)   \
      RB_COLOR(oright, field) = RB_BLACK;\
     RB_COLOR(tmp, field) = RB_RED; \
     RB_ROTATE_LEFT(head, tmp, oright, field);\
     tmp = RB_LEFT(parent, field); \
    }     \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_LEFT(tmp, field))  \
     RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    elm = RB_ROOT(head);   \
    break;     \
   }      \
  }       \
 }        \
 if
 (elm)       \
  RB_COLOR(elm, field) = RB_BLACK;   \
}         \
         \
attr struct type *       \
name##_RB_REMOVE(struct name *head, struct type *elm)   \
{         \
 struct type *child, *parent, *old = elm;   \
 int color;       \
 if (RB_LEFT(elm, field) == NULL)    \
  child = RB_RIGHT(elm, field);    \
 else if (RB_RIGHT(elm, field) == NULL)    \
  child = RB_LEFT(elm, field);    \
 else {        \
  struct type *left;     \
  elm = RB_RIGHT(elm, field);    \
  while ((left = RB_LEFT(elm, field)) != NULL)  \
   elm = left;     \
  child = RB_RIGHT(elm, field);    \
  parent = RB_PARENT(elm, field);    \
  color = RB_COLOR(elm, field);    \
  if (child)      \
   RB_PARENT(child, field) = parent;  \
  if (parent) {      \
   if (RB_LEFT(parent, field) == elm)  \
    RB_LEFT(parent, field) = child;  \
   else      \
```

```
    RB_RIGHT(parent, field) = child; \
   RB_AUGMENT(parent);    \
  } else       \
   RB_ROOT(head)
 = child;    \
  if (RB_PARENT(elm, field) == old)   \
   parent = elm;     \
  (elm)->field = (old)->field;    \
  if (RB_PARENT(old, field)) {    \
   if (RB_LEFT(RB_PARENT(old, field), field) == old)\
    RB_LEFT(RB_PARENT(old, field), field) = elm;\
   else      \
    RB_RIGHT(RB_PARENT(old, field), field) = elm;\
   RB_AUGMENT(RB_PARENT(old, field));  \
  } else       \
   RB_ROOT(head) = elm;    \
  RB_PARENT(RB_LEFT(old, field), field) = elm;  \
  if (RB_RIGHT(old, field))    \
   RB_PARENT(RB_RIGHT(old, field), field) = elm; \
  if (parent) {      \
   left = parent;     \
   do {      \
    RB_AUGMENT(left);   \
   } while ((left = RB_PARENT(left, field)) != NULL); \
  }       \
  goto color;      \
 }        \
 parent = RB_PARENT(elm, field);     \
 color = RB_COLOR(elm, field);     \
 if (child)       \
  RB_PARENT(child, field) = parent;   \
 if (parent) {       \
  if (RB_LEFT(parent, field) == elm)   \
   RB_LEFT(parent, field) = child;   \
  else       \
   RB_RIGHT(parent,
 field) = child;  \
  RB_AUGMENT(parent);     \
  } else       \
  RB_ROOT(head) = child;     \
color:         \
 if (color == RB_BLACK)      \
  name##_RB_REMOVE_COLOR(head, parent, child);  \
 return (old);       \
}         \
        \
/* Inserts a node into the RB tree */     \
attr struct type *       \
```

```
name##_RB_INSERT(struct name *head, struct type *elm)   \
{       \
 struct type *tmp;      \
 struct type *parent = NULL;     \
 int comp = 0;       \
 tmp = RB_ROOT(head);      \
 while (tmp) {       \
  parent = tmp;      \
  comp = (cmp)(elm, parent);    \
  if (comp < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else if (comp > 0)     \
   tmp = RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }       \
 RB_SET(elm, parent, field);     \
 if (parent != NULL) {      \
  if (comp < 0)      \
   RB_LEFT(parent, field) = elm;   \
  else       \
   RB_RIGHT(parent, field) = elm;   \
  RB_AUGMENT(parent);     \
 } else       \
  RB_ROOT(head) = elm;     \
 name##_RB_INSERT_COLOR(head,
 elm);    \
 return (NULL);       \
}       \
       \
/* Finds the node with the same key as elm */    \
attr struct type *       \
name##_RB_FIND(struct name *head, struct type *elm)   \
{       \
 struct type *tmp = RB_ROOT(head);    \
 int comp;       \
 while (tmp) {       \
  comp = cmp(elm, tmp);     \
  if (comp < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else if (comp > 0)     \
   tmp = RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }       \
 return (NULL);       \
}       \
       \
```

```
/* Finds the first node greater than or equal to the search key */ \
attr struct type *      \
name##_RB_NFIND(struct name *head, struct type *elm)   \
{       \
 struct type *tmp = RB_ROOT(head);    \
 struct type *res = NULL;     \
 int comp;       \
 while (tmp) {       \
  comp = cmp(elm, tmp);     \
  if (comp < 0) {      \
   res = tmp;     \
   tmp = RB_LEFT(tmp, field);   \
  }       \
  else if (comp > 0)     \
   tmp =
 RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }       \
 return (res);       \
}       \
       \
/* ARGSUSED */       \
attr struct type *      \
name##_RB_NEXT(struct type *elm)     \
{       \
 if (RB_RIGHT(elm, field)) {     \
  elm = RB_RIGHT(elm, field);    \
  while (RB_LEFT(elm, field))    \
   elm = RB_LEFT(elm, field);   \
 } else {       \
  if (RB_PARENT(elm, field) &&    \
      (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
   elm = RB_PARENT(elm, field);   \
  else {       \
   while (RB_PARENT(elm, field) &&   \
       (elm == RB_RIGHT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field);  \
   elm = RB_PARENT(elm, field);   \
  }       \
 }       \
 return (elm);       \
}       \
       \
/* ARGSUSED */       \
attr struct type *      \
name##_RB_PREV(struct type *elm)     \
{       \
```

```
 if (RB_LEFT(elm, field)) {     \
  elm = RB_LEFT(elm, field);    \
  while (RB_RIGHT(elm, field))    \
   elm = RB_RIGHT(elm, field);   \
 }
 else {       \
  if (RB_PARENT(elm, field) &&    \
      (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
   elm = RB_PARENT(elm, field);   \
  else {       \
   while (RB_PARENT(elm, field) &&   \
       (elm == RB_LEFT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field);  \
   elm = RB_PARENT(elm, field);   \
  }       \
 }       \
 return (elm);       \
}        \
        \
attr struct type *       \
name##_RB_MINMAX(struct name *head, int val)    \
{        \
 struct type *tmp = RB_ROOT(head);    \
 struct type *parent = NULL;     \
 while (tmp) {       \
  parent = tmp;      \
  if (val < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else       \
   tmp = RB_RIGHT(tmp, field);   \
 }       \
 return (parent);      \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
 y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x)  name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x)  name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head)     \
```

```
 for ((x) = RB_MIN(name, head);     \
      (x) != NULL;      \
      (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y)     \
 for ((x) = (y);       \
     ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y)    \
 for ((x) = RB_MIN(name, head);     \
     ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head)    \
 for ((x) = RB_MAX(name, head);     \
      (x) != NULL;      \
      (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y)    \
 for ((x) = (y);       \
     ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,
 head, y)   \
 for ((x) = RB_MAX(name, head);     \
     ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
      (x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-$_kernver.tar.xz
 revert-broken-uapi.patch
 0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch
```

1.36 ca-certificates 20250911-r0

1.36.1 Available under license :

No license file was found, but licenses were detected in source scan.

/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-certificates/mozilla/HARICA_TLS_RSA_Root_CA_2021.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-certificates/mozilla/QuoVadis_Root_CA_2.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-certificates/mozilla/TWCA_Global_Root_CA.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-certificates/mozilla/TrustAsia_TLS_RSA_Root_CA.crt-----BEGIN CERTIFICATE-----

Found in path(s):
*
No license file was found, but licenses were detected in source scan.

-----BEGIN CERTIFICATE-----
MIIFWjCCA0KgAwIBAgISEdK7udcjGJ5AXwqdLdDfJWfRMA0GCSqGSIb3DQEBDAUA
MEYxCzAJBgNVBAYTAkJFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMRwwGgYD
VQQDExNHbG9iYWxTaWduIFJvb3QgUjQ2MB4XDTE5MDMyMDAwMDAwMFoXDTQ2MDMy
MDAwMDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYt
c2ExHDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBSNDYwggIiMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCsrHQy6LNl5brtQyYdpokNRbopiLKkHWPd08EsCVeJ
OaFV6Wc0dwxu5FUdUiXSE2te4R2pt32JMl8Nnp8semNgQB+msLZ4j5lUlghYruQG
vGIFAha/r6gjA7aUD7xubMLL1aa7DOn2wQL7Id5m3RerdELv8HQvJfTqa1VbkNud
316HCkD7rRlr+/fKYIje2sGP1q7Vf9Q8g+7XFkyDRTNrJ9CG0Bwta/OrffGFqfUo
0q3v84RLHIf8E6M6cqJaESvWJ3En7YEtbWaBkoe0G1h6zD8K+kZPTXhc+CtI4wSE
y132tGqzZfxCnlEmIyDLPRT5ge1lFgBPGmSXZgjPjHvjK8Cd+RTyG/FWaha/LIWF
zXg4mutCagI0GIMXTpRW+LaCtfOW3T3zvn8gdz57GSNrLNRyc0NXfeD412lPFzYE
+cCQYDdF3uYM2HSNrpyibXRdQr4G9dlkbgIQrImwTDsHTUB+JMWKmIJ5jqSngiCN
I/onccnfxkF0oE32kRbcRoxfKWMxWXEM2G/CtjJ9++ZdU6Z+Ffy7dXxd7Pj2Fxzs
x2sZy/N78CsHpdlseVR2bJ0cpm4O6XkMqCNqo98bMDGfsVR7/mrLZqrcZdCinkqa
ByFrgY/bxFn63iLABJzjqls2k+g9vXqhnQt2sQvHnf3PmKgGwvgqo6GDoLclcqUC
4wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNV
HQ4EFgQUA1yrc4GHqMywptWU4jaWSf8FmSwwDQYJKoZIhvcNAQEMBQADggIBAHx4
7PYCLLtbfpIrXTncvtgdokIzTfnvpCo7RGkerNlFo048p9gkUbJUHJNOxO97k4Vg
JuoJSOD1u8fpaNK7ajFxzHmuEajwmf3lH7wvqMxX63bEIaZHU1VNaL8FpO7XJqti
2kM3S+LGteWygxk6x9PbTZ4IevPuzz5i+6zoYMzRx6Fcg0XERczzF2sUyQQCPtIk
pnnpHs6i58FZFZ8d4kuaPp92CC1r2LpXFNqD6v6MVenQTqnMdzGxRBF6XLE+0xRF
FRhiJBPSy03OXIPBNvIQtQ6IbbjhVp+J3pZmOUdkLG5NrmJ7v2B0GbhWrJKsFjLt
rWhV/pi60zTe9Mlhww6G9kuEYO4Ne7UyWHmRVSyBQ7N0H3qqJZ4d16GLuc1CLgSk
ZoNNiTW2bKg2SnkheCLQQrzRQDGQob4Ez8pn7fXwgNNgyYMqIgXQBztSvwyeqiv5
u+YfjyW6hY0XHgL+XVAEV8/+LbzvXMAaq7afJMbfc2hIkCwU9D9SGuTSyxTDYWnP
4vkYxboznxSjBF25cfe1lNj2M8FawTSLfJvdkzrnE6JwYZ+vj+vYxXX4M2bUdGc6

N3ec592kD3ZDZopD8p/7DEJ4Y9HiD2971KE9dJeFt0g5QdYg/NA6s/rob8SKunE3
vouXsXgxT7PntgMTzlSdriVZzH81Xwj3QEUxeCp6
-----END
CERTIFICATE-----

Found in path(s):
*
/ca-certificates-20250911-r0-tar-gz/usr/share/ca-certificates/mozilla/GlobalSign_Root_R46.crt

1.37 aws-sdk-go-v2 1.39.4

1.37.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
 // This member is required.
 Issuer *types.Issuer

 // License name.
 //
 // This member is required.
 LicenseName *string

 // Product name.
 //
 // This member is required.
 ProductName *string

 // Product SKU.
 //
```

```
 // This member is required.
 ProductSKU *string

 // Date and time range during which the license is valid, in ISO8601-UTC format.
 //
 // This member is required.
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
```

```
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
```

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
```

```
//   - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
```

```
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
```

```
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
```

```
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
```

```
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
          }

 },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

          {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "ServerType": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
               "traits": {
                  "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                  "smithy.api#required": {}

   }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
         "type": "structure",
         "members": {
            "LicenseServerEndpoint": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
               "traits": {
                  "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
```

```
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
```

```
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members":
{
            "Attribute": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
              }
            },
            "Operation": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
              }
            },
            "Value": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
            }

          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
```

```
 contains the user information for the product subscription.</p>"
                }
            },
            "AssociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
                }
            },
            "DisassociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
```

```
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
```

```
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
```

```
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
```

```
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },

"properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                 "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

```
true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

"fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      }
                    ]
                  },

                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
```

```
                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }

            ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [

        {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
```

```
                    "argv": [

                {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
                },
                true
            ]
            }
        ],
        "rules": [
            {
            "conditions": [],
            "endpoint": {

    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
            }
        ],
        "type": "tree"
        },
        {
        "conditions": [],
        "error": "FIPS is enabled but this partition does not support FIPS",
        "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
            {
            "fn": "booleanEquals",
            "argv": [
                {
                "ref": "UseDualStack"
                },
                true
            ]
            }
            ],
            "rules": [
```

```
                {

        "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]

    }
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }

                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "DualStack is enabled but this partition does not support DualStack",
                  "type": "error"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
```

```
                    "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",

        "UseFIPS": true,
              "UseDualStack": false
            }
```

```
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

  "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
 true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
```

```
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
```

```
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
```

```
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

      "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
                },
                {
                  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://example.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,

        "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://example.com"
                    }
                  },
                  "params": {
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                  "expect": {

   "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
```

```
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
            "type": "structure",
            "members": {
              "ProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                  "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
                }
              },
              "HealthStatus": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                  "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
              },
              "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
              },
              "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>"
            }
          },
          "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
            }
          },
          "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
```

```
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
```

```
            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
              "name": "DELETED",

"value": "DELETED",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
```

```
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
              }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
                "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "IdentityProviderSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
         "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources that contain details about the Active Directory identity providers that meet the request criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
```

```
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results
```

```
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

     "members": {
            "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
     "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
```

```
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

      {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
 {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
```

```
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
```

```
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
 empty, there are additional elements that the service hasn't included in this request. Use this token with the next
 request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}

 }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
```

```
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
 {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
              }
            },
            "SubscriptionStartDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
              }
            },
            "SubscriptionEndDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
```

reference to the credentials provider that's used for RDS license server user administration. ",
"smithy.api#required": {}
}
}
},
"traits": {

"smithy.api#documentation": " Server settings that are specific to a Remote Desktop Services (RDS) license server. "
}
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
"type": "operation",
"input": {
"target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
},
"errors": [
{
"target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
"smithy.api#documentation": " Registers an identity provider for user-based subscriptions. ",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/identity-provider/RegisterIdentityProvider"

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
                }

            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
              "smithy.api#length": {
                "min": 1
              }
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
```

```
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }

    },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC. "
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " Starts a product subscription for a user with the specified identity provider. <note> Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing. For more information, see Viewing your monthly charges in the Amazon Web Services Billing User Guide. </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

```
 "uri": "/user/StartProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
            }
          }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
      "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
          "smithy.api#error": "client"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {

         "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

        }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
          "smithy.api#http": {

    "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
```

```
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
```

```
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}
```

```
type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if
```

```
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
```

```
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
```

```
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
```

```
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
```

```
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod
```

```
// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
```

```
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
 middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
```

```
//
// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
```

```
c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
```

```
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
```

```
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
```

```
 OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
```

InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
```

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
```

```
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
            "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
```

```
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"

          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
```

```
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
           "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
```

```
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
        "Region": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
```

```
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

     "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
            }
        ],
        "traits": {
            "aws.api#service": {
                "sdkId": "License Manager Linux Subscriptions",
                 "arnNamespace": "license-manager-linux-subscriptions",
                "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
            },
            "aws.auth#sigv4": {
                "name": "license-manager-linux-subscriptions"
            },
            "aws.protocols#restJson1": {},
            "smithy.api#cors": {
                "additionalAllowedHeaders": [
                    "*",
                    "content-type",
                    "x-amz-content-sha256",
                    "x-amz-user-agent",
                    "x-amzn-platform-id",
                    "x-amzn-trace-id"
                ],
                "additionalExposedHeaders": [
                    "x-amzn-errortype",
                    "x-amzn-requestid",
                    "x-amzn-trace-id"
                ]
            },
            "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager Linux Subscriptions",
            "smithy.rules#endpointRuleSet": {
                "version": "1.0",
                "parameters": {
                    "Region": {
                        "builtIn": "AWS::Region",
                        "required": false,
                        "documentation": "The AWS region used to dispatch the request.",
                        "type": "String"
                    },
                    "UseDualStack": {
```

```
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
             "type": "Boolean"
          },
          "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
          },
          "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
          }
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
```

```
        ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
            },
            {
              "conditions": [],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]

}
                  ],
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "endpoint": {
                    "url": {
                      "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ],
              "type": "tree"

          }
          ],
          "type": "tree"
        },
        {
          "conditions": [],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
```

```
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

        "fn": "aws.partition",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ],
                    "assign": "PartitionResult"
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {

                       "ref": "UseFIPS"
                          },
                          true
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }

    ],
                    "rules": [
                      {
                        "conditions": [
```

```
                                        {
                                          "fn": "booleanEquals",
                                          "argv": [
                                            true,
                                            {
                                              "fn": "getAttr",
                                              "argv": [
                                                {
                                                  "ref": "PartitionResult"

                    },
                                                "supportsFIPS"
                                              ]
                                            }
                                          ]
                                        },
                                        {
                                          "fn": "booleanEquals",
                                          "argv": [
                                            true,
                                            {
                                              "fn": "getAttr",
                                              "argv": [

                                               {
                                                  "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                              ]
                                            }
                                          ]
                                        }
                                      ],
                                      "rules": [
                                        {
                                          "conditions": [],

        "rules": [
                                            {
                                              "conditions": [],
                                              "endpoint": {
                                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                              },
                                              "type": "endpoint"
                                            }
```

```
                ],

            "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        }
    ],
    "type": "tree"
},
{

"conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {

            "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
```

```
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "rules": [
                                        {
                                            "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }
                                    ],
                                    "type": "tree"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS is enabled
 but this partition does not support FIPS",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true

                ]
                        }
                    ],
                    "rules": [
                        {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {

                          "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [

        {
                          "conditions": [],
                          "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
                      ],
                      "type": "tree"

                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "DualStack is enabled but this partition does not support DualStack",
                "type": "error"
              }
            ],
            "type": "tree"
          },
```

```
                                {
                                    "conditions": [],
                                    "rules": [
                                        {
                                            "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }
                                    ],
                                    "type": "tree"
                                }
                            ],
                            "type": "tree"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "Invalid Configuration: Missing Region",
                    "type": "error"
                }
            ],
            "type": "tree"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

        }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
```

```
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }

  },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
```

```
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
       },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
```

```
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": true,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
                      }

                   },
                    "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {

  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
                      }
                    },
```

```
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
```

```
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },

    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
```

```
      "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
               }
            },
            {
               "documentation": "For custom endpoint with fips disabled and dualstack enabled",
               "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
               },
               "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
               }
            },
            {
               "documentation": "Missing region",

            "expect": {
                  "error": "Invalid Configuration: Missing Region"
               }
            }
         ],
         "version": "1.0"
      }
   }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
   "type": "string",
   "traits": {
      "smithy.api#enum": [
         {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled LinuxSubscriptionsDiscovery"
         },
         {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
         }
      ]
   }
},
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
          "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists
```

```
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "Instances"
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
         "type": "structure",
         "members": {
            "Filters": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n           <li>\n              <p>\n                 <code>AccountID</code>\n
</p>\n           </li>\n           <li>\n              <p>\n                 <code>AmiID</code>\n              </p>\n           </li>\n
        <li>\n              <p>\n                 <code>DualSubscription</code>\n              </p>\n           </li>\n
<li>\n              <p>\n                 <code>InstanceID</code>\n              </p>\n           </li>\n           <li>\n
  <p>\n                 <code>InstanceType</code>\n              </p>\n           </li>\n           <li>\n              <p>\n
    <code>ProductCode</code>\n              </p>\n           </li>\n           <li>\n              <p>\n
<code>Region</code>\n              </p>\n           </li>\n           <li>\n              <p>\n
<code>Status</code>\n              </p>\n           </li>\n           <li>\n              <p>\n
<code>UsageOperation</code>\n              </p>\n           </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n              <p>\n                 <code>contains</code>\n              </p>\n           </li>\n           <li>\n
<p>\n                 <code>equals</code>\n              </p>\n           </li>\n           <li>\n
              <p>\n                 <code>Notequal</code>\n              </p>\n           </li>\n        </ul>"
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
               }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                     "min": 1,
                     "max": 16384
                  }
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
         "type": "structure",
         "members": {
            "Subscriptions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
               "traits": {
                  "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

               }
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
         "type": "operation",
         "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
         },
         "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
         },
```

```
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
               }
            }
         },

         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
         "type": "structure",
         "members": {
            "RegisteredSubscriptionProviders": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
               "traits": {
                  "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
         ],
         "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
            "smithy.api#http": {
               "method": "GET",
               "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
         "type": "structure",
         "members": {
            "resourceArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                  "smithy.api#httpLabel": {},
                  "smithy.api#required": {}

               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
         "type": "structure",
         "members": {
            "tags": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {

"name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
```

```
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },

         {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
              "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
              "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
              }
            },
            "SubscriptionProviderArn": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
```

```
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
                }
            },
             "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
                }
            },
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
                }
            },
            "LastSuccessfulDataRetrievalTime": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
                }
            }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
```

```
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }
        }
       },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
```

```
            "Type": {
               "target": "smithy.api#String",
               "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with
Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace.
Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing
from Amazon EC2 or BYOS.</p>"
               }
            },
            "InstanceCount": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
               "traits": {
                  "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
         "type": "list",

       "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "RedHat",
                  "value": "RedHat",
                  "documentation": "RedHat subscription provider namespace"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
```

```
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                 "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
      "type": "map",
      "key":
{
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
       },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
               "method": "DELETE",
               "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
         "type": "structure",

    "members": {
            "resourceArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                  "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            },
            "tagKeys": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
               "traits": {
                  "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                  "smithy.api#httpQuery": "tagKeys",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input":
 {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
         "type": "structure",
         "members": {},
         "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
```

```
from.</p>",
                    "smithy.api#required": {}
                }
            },
            "AllowUpdate": {
                "target": "smithy.api#Boolean",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
                }
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
            "HomeRegions": {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.38 aws-sdk-go-v2-service-sts 1.38.9

1.38.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
```

```
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange
```

```
 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step

```
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
```

```
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
```

```
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
```

```
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
```

```
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
```

```
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
            "DomainName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
                }
            },
            "DomainIpv4List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainIpv6List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
```

```
array of subnets that apply for the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                     "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
```

```
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

                 "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                 "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }

  },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

       {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
```

```
license server.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
          "type": "structure",
          "members":
{
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
              }
            },
            "LicenseServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                    "smithy.api#required": {}

  }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
```

```
 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
```

```
  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
               }
            },
            "InstanceUserArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
```

```
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

                 "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }

            },
            "FailureMessage": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
        },
        "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
      "type": "structure",
      "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        },
        "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                "smithy.api#required": {}
            }
        },
```

```
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
          },
          "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
```

```
            },
            "DisassociationDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
         "type": "list",
         "member": {

     "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
            },
            {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
      },
      {

        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id",
          "content-length",
           "x-api-key",
          "authorization",
          "x-amz-date",
          "x-amz-security-token",
          "Access-Control-Allow-Headers",
          "Access-Control-Allow-Methods",
          "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
          "x-amzn-errortype",
```

```
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
```

```
                    "ref": "Endpoint"
                }
            ]

    }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [

{
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },

        "properties": {},
                        "headers": {}
                    },
```

```
                "type": "endpoint"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "aws.partition",
                     "argv": [
                       {
                         "ref": "Region"
                       }
                    ],
                    "assign": "PartitionResult"
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },

true
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
```

```
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

"fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    },

                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
```

```
                "headers": {}
              },
              "type": "endpoint"
            }

        ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
            "type": "error"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [

    {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "fn": "getAttr",
                    "argv": [

          {
                        "ref": "PartitionResult"
                      },
                      "supportsFIPS"
                    ]
                  },
                  true
                ]
```

```
                }
              ],
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "FIPS is enabled but this partition does not support FIPS",
              "type": "error"

}
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {

    "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
```

```
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]

                }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    }
],
```

```
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",

             "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

    "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
 true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
```

```
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

  "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
```

```
                "UseFIPS": false,

        "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
               }
            },
            {
              "documentation": "Missing region",
              "expect": {
```

```
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
     }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
```

```
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
```

```
        },
        "LicenseServers": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
          }
        },
        "CreationTime": {
          "target": "smithy.api#Timestamp",

    "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
```

```
                "value": "PROVISIONED",
                "documentation": ""
              },
              {
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

      {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
```

```
operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

},
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
```

```
            "ResourceArn": {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                  "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
         "type": "structure",
         "members": {
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}

 }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
```

```
    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }

 },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output":
 {}
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS) license server.</p>"
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
```

```
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

          },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                  "smithy.api#required": {}
                }
              }
          },
          "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
              "smithy.api#length": {
                "min": 1
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
 {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
```

```
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
```

```
{
        "type": "structure",
        "members": {
           "message": {
              "target": "smithy.api#String"
           }
        },
        "traits": {
           "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
           "smithy.api#error": "client"
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
           "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                 "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                 "smithy.api#length": {
                    "min": 1
                 },
                 "smithy.api#required": {}
              }

     },
           "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                 "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                 "smithy.api#required": {}
              }
           }
        },
        "traits": {
           "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
           "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
```

```
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        },
        "IdentityProvider":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
           }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }

        }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {

    "min": 0,
            "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

            }
            },
            "TagKeys": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
        "smithy.api#http": {

  "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
```

```
            },
            "IdentityProviderArn":
 {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }

            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
```

```
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
```

```
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
```

```
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string
```

```
 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
```

```
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}
```

```
type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
```

```
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string
```

```
 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
```

```
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
```

```
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
```

```
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
           "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
           },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
```

```
        "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
```

```
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
```

```
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
```

```
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }

        ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
```

```
                {
                  "conditions": [
                    {

            "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
                      "assign": "PartitionResult"
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {

                          "ref": "UseFIPS"
                            },
                            true
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }

],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
```

```
                                        "ref": "PartitionResult"

                    },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [

                                     {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {

      "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

            "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled
but this partition does not support FIPS",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
```

```
                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

        {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
"testCases": [
{
"documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
"expect": {
"endpoint": {
"url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

}
},
"params": {
"Region": "us-east-1",
"UseFIPS": true,
"UseDualStack": true
}
},
{
"documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
"expect": {
"endpoint": {
"url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
}
},
"params": {
"Region": "us-east-1",
```

```
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                }

    },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
```

```
"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                      }
                     },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
              "Region": "us-isob-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",

                  "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
            "type": "string",
            "traits": {
              "smithy.api#enum": [
                {
                  "name": "Enabled",
                  "value": "Enabled",
                  "documentation": "Enabled LinuxSubscriptionsDiscovery"
                },
                {
                  "name": "Disabled",
                  "value": "Disabled",
                  "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
              ]
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
            "type": "structure",
            "members": {
              "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                  "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
```

```
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
```

```
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
```

" Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations. ",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
"inputToken": "NextToken",
"outputToken": "NextToken",
"pageSize": "MaxResults",
"items": "Subscriptions"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
"type": "structure",
"members": {
"Filters": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

"traits": {
"smithy.api#documentation": " An array of structures that you can use to filter the results to those that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n <code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't match a certain subscription's name. \n The valid names for this filter are: \n \n \n \n <code>Subscription</code>\n \n \n \n The valid Operators for this filter are: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
}
},
"MaxResults": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
"traits": {
"smithy.api#documentation": " The maximum items to return in a request. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
"smithy.api#length": {
"min": 1,
"max": 16384

```
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

     "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
 return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
 nextToken from a previously truncated response.</p>"
              }
            }
          },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
```

```
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
          "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
                }
            },
            "SubscriptionProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
```

```
        "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
                "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
        },
        "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
            }
        },
        "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
        },
        "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
        },
        "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

                "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
               },
              {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
              },
              {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
              },
              {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
              }
            ]
```

```
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
                }
```

```
            },
            "InstanceCount": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
               "traits": {
                  "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
         "type": "list",

       "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "RedHat",
                  "value": "RedHat",
                  "documentation": "RedHat subscription provider namespace"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
         "type":
 "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
```

```
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
 {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
```

```
Region.</p>"
                }
            }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }

    },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
```

```
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.39 aws-sdk-go-v2-config 1.31.15

1.39.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
```

```
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag
```

```
 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
```

```
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
```

```
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
```

```
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength

```
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
```

```
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense
```

```
 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
```

```
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
```

```
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
```

```
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {
```

```
 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
```

```
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
```

ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
        "type": "structure",
        "members": {
            "DirectoryId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
                "traits": {
                    "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
                }
            },
            "ActiveDirectorySettings":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
                }
            },
            "ActiveDirectoryType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
                }
            },
            "IsSharedActiveDirectory": {
                "target": "smithy.api#Boolean",
                "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
```

```
            "DomainName": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
              }
            },
            "DomainIpv4List": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainIpv6List": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
              }
            },
            "DomainNetworkSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
             "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {

     "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider for the user.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
```

```
state of the resource.</p>",
            "smithy.api#error": "server"
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
{
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
```

```
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
              "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
              "smithy.api#idempotent": {}
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
            "type": "structure",
            "members": {
              "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                  "smithy.api#required": {}
                }
              },
              "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                  "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                  "smithy.api#required": {}

    }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
            "type": "structure",
            "members": {
              "LicenseServerEndpoint": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
          "smithy.api#http": {
```

```
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
```

```
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
```

```
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
                 }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
                }
            }
        },
        "traits": {
```

```
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },

    "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
            "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
            },
```

```
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Value of the filter.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
            "ActiveDirectoryIdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
```

about an identity provider. ",
"smithy.api#required": {}
}
},
"Settings": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
"traits": {
"smithy.api#documentation": " The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints. ",

"smithy.api#required": {}
}
},
"Product": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the user-based subscription product. ",
"smithy.api#required": {}
}
},
"Status": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The status of the identity provider. ",
"smithy.api#required": {}
}
},
"IdentityProviderArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The Amazon Resource Name (ARN) of the identity provider. "
}

},
"FailureMessage": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The failure message associated with an identity provider. "
}
},
"OwnerAccountId": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The AWS Account ID of the owner of this resource. "
}
}
},

```
        "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
```

```
            },
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                  "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                  "smithy.api#required": {}
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
```

```
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
                }
            },
            "AssociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
                }
            },
            "DisassociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
```

```
"type": "service",
"version": "2018-05-10",
"operations": [
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
```

```
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
        },
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
```

```
    "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
```

```
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },

            "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
```

```
                    "ref": "Region"
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                     "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },

true
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {

"fn": "booleanEquals",
                                    "argv": [
```

```
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                      ]
                    }
                  ]
                },

                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                      ]
                    }

              ]
                }
              ],
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }

        ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "FIPS and DualStack are enabled, but this partition does not support one or
```

```
both",
                                    "type": "error"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [

        {
                                            "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [
                                        {
                                            "fn": "booleanEquals",
                                            "argv": [
                                                {
                                                    "fn": "getAttr",
                                                    "argv": [

                                {
                                                        "ref": "PartitionResult"
                                                    },
                                                    "supportsFIPS"
                                                ]
                                            },
                                            true
                                        ]
                                    }
                                ],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {

                    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
```

```
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"
                  },
                  {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"

}
                ],
                "type": "tree"
              },
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {

    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsDualStack"
                            ]

     }
                      ]
                    }
                  ],
                  "rules": [
```

```
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }

                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
```

```
                {
                  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",

              "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                  }
```

```
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
```

```
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
```

```
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
```

```
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

 "UseDualStack": false,
          "Endpoint": "https://example.com"
        }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
```

```
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
```

```
            "ProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
                }
            },
            "HealthStatus": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The type of license server.</p>"
            }
        },
        "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
            }
        },
        "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
        },
        "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
        },
        "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
        },
        "CreationTime": {
            "target": "smithy.api#Timestamp",
```

```
    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
```

```
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
```

```
            },
            "ServerSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                  "smithy.api#required": {}
               }

  }
         },
         "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
```

```
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
               "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
               }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources that contain details about the Active Directory identity providers that meet the request criteria.</p>",
                "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
```

```
      "type": "structure",

    "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
```

```
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                 "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }

 },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output":
 {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
          }
        },
```

```
            "SubscriptionEndDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
```

```
            }

        },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
```

```
            "message": {
               "target": "smithy.api#String"

            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
         "type": "structure",
         "members": {
            "SecretId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                  "smithy.api#length": {
                     "min": 1
                  }
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
         "type": "string",
         "traits": {
            "smithy.api#length": {
               "min": 5,
               "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
         "type": "structure",
         "members": {
            "Endpoint": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
               }
```

```
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
          "type": "union",

         "members": {
            "RdsSalSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "RDS_SAL",
                "value": "RDS_SAL",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
 {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }

    },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
      "type": "operation",
      "input":
{
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

         {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
```

```
            "target": "smithy.api#String",
             "traits": {
               "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
               "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
               "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {

"min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
```

```
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {

           "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
          "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

          }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
                }
            },
            "UpdateSettings": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
```

```
            },
            "SecurityGroupId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
               "traits": {
                  "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
               }
            }
         },

         "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
```

```
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
```

```
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region:
```

```
        region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
```

```
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string
```

```
// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
```

```
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {
```

```
// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
```

```
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options)) (*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}
```

```
type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {
```

```
 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
```

```
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
```

```
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
```

```
    {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/DeregisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
```

```
            }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
```

```
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
               "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
        "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
        },
```

```
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
```

```
"type": "service",
"version": "2018-05-10",
"operations": [
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
         "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
```

```
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "Boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
              }
```

```
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }

    ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]

}
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
```

```
            },
            {
              "conditions": [],
              "endpoint": {
                "url": {
                  "ref": "Endpoint"
                },
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ],
          "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {

    "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
```

```
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {

                                "ref": "UseFIPS"
                                },
                                true
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }

],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"

},
                                                "supportsFIPS"
                                            ]
                                        }
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
```

```
                "argv": [

                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
                ]
            }
            ]
        }
        ],
        "rules": [
        {
            "conditions": [],

    "rules": [
                {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
                }
            ],

            "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

            "type": "error"
        }
        ],
        "type": "tree"
    },
    {

        "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
```

```
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {

                    "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "fn": "getAttr",
                                                "argv": [
                                                    {
                                                        "ref": "PartitionResult"
                                                    },
                                                    "supportsFIPS"
                                                ]
                                            },
                                            true

                                        ]
                                    }
                                ],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "rules": [
                                            {
                                                "conditions": [],
                                                "endpoint": {
                                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                                    "headers": {}
                                                },
                                                "type": "endpoint"
                                            }
                                        ],
                                        "type": "tree"
                                    }
                                ],
                                "type": "tree"
```

```
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled
but this partition does not support FIPS",
                    "type": "error"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true

    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {

                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [
```

```
        {
                                                "conditions": [],
                                                "endpoint": {
                                                   "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                   "properties": {},
                                                   "headers": {}
                                                },
                                                "type": "endpoint"
                                             }
                                          ],
                                          "type": "tree"

                                    }
                                       ],
                                       "type": "tree"
                                    },
                                    {
                                       "conditions": [],
                                       "error": "DualStack is enabled but this partition does not support DualStack",
                                       "type": "error"
                                    }
                                 ],
                                 "type": "tree"
                              },
                              {
                                 "conditions": [],
                                 "rules": [
                                    {
                                       "conditions": [],
                                       "endpoint": {
                                          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                       },
                                       "type": "endpoint"
                                    }
                                 ],
                                 "type": "tree"
                              }
                           ],
                           "type": "tree"
                        }
                     ],
                     "type": "tree"
                  },
                  {
```

```
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ],
            "type": "tree"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

        }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {

         "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

      },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

      },
      "params": {
        "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

    },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
```

```
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "Missing region",
```

```
                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
```

```
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
```

```
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
          "type": "structure",
          "members": {
            "Instances": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
              "traits": {
```

```
                "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
 objects.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                  }
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
            "type": "structure",
            "members": {
              "Subscriptions": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
```

```
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
```

```
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
           }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "EQUAL",
              "value": "Equal",
              "documentation": "Equal operator"
            },
            {
              "name": "NOT_EQUAL",
              "value": "NotEqual",
              "documentation": "Not equal operator"
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
```

```
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {

                    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
```

```
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
          }
        },
        "SubscriptionProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
```

```
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
```

```
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
```

```
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with
Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace.
Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing
from Amazon EC2 or BYOS.</p>"
                }
            },
            "InstanceCount": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
                "traits": {
                    "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
```

```
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
```

```
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

   "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
{}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
```

```
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
            }

    },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
            }
           }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
          "smithy.api#error": "client"
        }
      }
    }
```

}

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.40 libssl3 3.5.4-r0

1.40.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under
the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

a) the GNU General Public License as published by the Free
Software Foundation; either version 1, or (at your option) any
later version, or
b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so

that any problems introduced by others will not reflect on the original
authors' reputations.

The precise terms and conditions
for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which
contains a notice placed by the copyright holder saying it may be
distributed under the terms of this General Public License. The
"Program", below, refers to any such program or work, and a "work based
on the Program" means either the Program or any work containing the
Program or a portion of it, either verbatim or with modifications. Each
licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source
code as you receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice and
disclaimer of warranty; keep intact all the notices that refer to this
General Public License and to the absence of any warranty; and give any
other recipients of the
Program a copy of this General Public License
along with the Program. You may charge a fee for the physical act of
transferring a copy.

2. You may modify your copy or copies of the Program or any portion of
it, and copy and distribute such modifications under the terms of Paragraph
1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that
you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that
in whole or in part contains the Program or any part thereof, either
with or without modifications, to be licensed at no charge to all
third parties under the terms of this General Public License (except
that you may choose to grant warranty protection to some or all
third parties, at your option).

c) If the modified program normally reads commands interactively when
run, you must cause it, when started running for such interactive
use
in the simplest and most usual way, to print or display an
announcement including an appropriate copyright notice and a notice
that there is no warranty (or else, saying that you provide a

warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties

remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free
Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter
the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright

Holder, and derivatives of that collection
of files created through
textual modification.
- "Standard Version" refers to such a Package if it has not been modified,
or has been modified in accordance with the wishes of the Copyright
Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for
the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media
cost, duplication charges, time of people involved, and so on. (You will
not be required to justify it to the Copyright Holder, but only to the
computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though
there may be fees involved in handling the item. It also means that
recipients of the item may redistribute it under the same conditions they
received it.

1. You may make and give away verbatim copies of
the source form of the
Standard Version of this Package without restriction, provided that you
duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived
from the Public Domain or from the Copyright Holder. A Package modified in such
a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that
you insert a prominent notice in each changed file stating how and when you
changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them
Freely Available, such as by posting said modifications to Usenet or an
equivalent medium, or placing the modifications on a major archive site
such as ftp.uu.net, or by allowing the Copyright Holder to include your
modifications in the Standard Version of the Package.

b) use the modified
Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with
standard executables, which must also be provided, and provide a separate
manual page for each non-standard executable that clearly documents how it
differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable

form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files,
together with instructions (in the manual page or equivalent) on where to
get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package
with your modifications.

c) accompany any non-standard executables with their corresponding Standard
Version executables, giving the non-standard executables non-standard
names,
and clearly documenting the differences in manual pages (or
equivalent), together with instructions on where to get the Standard
Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this
Package. You may charge any fee you choose for support of this Package. You
may not charge a fee for this Package itself. However, you may distribute this
Package in aggregate with other (possibly commercial) programs as part of a
larger (possibly commercial) software distribution provided that you do not
advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output
from the programs of this Package do not automatically fall under the copyright
of this Package, but belong to whomever generated them, and may be sold
commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package
shall not
be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote
products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.41 musl 1.2.5-r10

1.41.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* iconv.c
* Implementation of SUSv4 XCU iconv utility
* Copyright © 2011 Rich Felker
* Licensed under the terms of the GNU General Public License, v2 or later
*/

Found in path(s):
* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/iconv.c
No license file was found, but licenses were detected in source scan.

```
/*-
* Copyright (c) 1996, 1998 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by J.T. Conklin.
*
* Mostly rewritten to be used in Alpine Linux (with musl c-library)
* by Timo Teräs.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
 NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/getconf.c
No license file was found, but licenses were detected in source scan.

```
/*-
* Copyright (c) 2004-2006 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Luke Mewburn.
```

* Timo Teräs cleaned up the code for use in Alpine Linux with musl libc.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
*
 TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/getent.c

1.42 aws-sdk-go-v2-internal-ini 1.8.4

1.42.1 Available under license :

GetLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization

```
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
```

```
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus
```

```
// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
```

```
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
```

```
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
```

```
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

```
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
```

```
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
```

```
addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
```

DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
  InterceptBeforeDeserialization

```
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
```

```
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
```

```
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
```

```
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
```

```
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
```

RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
```

```
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
      "type": "structure",
      "members": {
        "DomainName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
          }
        },
        "DomainIpv4List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv4
```

```
addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainIpv6List": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
                }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "SELF_MANAGED",
                 "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
              },
              {
```

```
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
              }
            },
```

```
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
               "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
          }

 },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
          },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
```

```
Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
```

```
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
          "type": "structure",
          "members": {
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}

    }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
            "type": "structure",
            "members": {
              "LicenseServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
                "traits": {
                  "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
                }
              }
            },
            "traits": {
              "smithy.api#output": {}
            }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
```

```
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
```

```
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
```

```
                "smithy.api#required": {}
            }
          }
        },

    "traits": {
          "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
```

```
        "traits": {
            "aws.api#service": {
                "sdkId": "License Manager User Subscriptions",
                "arnNamespace": "license-manager-user-subscriptions",
                "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
            },
            "aws.auth#sigv4": {
                "name": "license-manager-user-subscriptions"
            },
            "aws.protocols#restJson1": {},
            "smithy.api#cors": {
                "additionalAllowedHeaders": [
                    "*",
                    "content-type",
                    "x-amz-content-sha256",
                    "x-amz-user-agent",
                    "x-amzn-platform-id",
                    "x-amzn-trace-id",
                    "content-length",
                     "x-api-key",
                    "authorization",
                    "x-amz-date",
                    "x-amz-security-token",
                    "Access-Control-Allow-Headers",
                    "Access-Control-Allow-Methods",
                    "Access-Control-Allow-Origin"
                ],
                "additionalExposedHeaders": [
                    "x-amzn-errortype",
                    "x-amzn-requestid",
                    "x-amzn-trace-id"
                ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager User Subscriptions",
            "smithy.rules#endpointRuleSet": {
                "version": "1.0",
                "parameters": {

        "Region": {
                        "builtIn": "AWS::Region",
                        "required": false,
                        "documentation": "The AWS region used to dispatch the request.",
                        "type": "string"
                    },
                    "UseDualStack": {
                        "builtIn": "AWS::UseDualStack",
```

```
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

}
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
```

```
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [

{
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },

        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
```

```
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {

"fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
```

```
                                ]
                            }
                        ]
                    },

                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }

                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }

        ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [

{
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "fn": "getAttr",
                          "argv": [

                {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {

          "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
```

```
                        "type": "error"

            }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {

        "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]

         }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
```

```
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
```

```
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",

  "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
```

```
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {

"endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
```

```
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
```

```
                {
                  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint":
{
                      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                  },
```

```
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,

         "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
    disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
```

```
                },
                {
                  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                  "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                   }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
          "type": "structure",
          "members": {
            "ProvisioningStatus": {
              "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
              }
            },
```

```
            "HealthStatus": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
```

```
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
                }
            }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
                    "value": "PROVISIONING_FAILED",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONED",
                    "value": "PROVISIONED",
                    "documentation": ""
                },
                {
                    "name": "DELETING",
                    "value": "DELETING",
                    "documentation": ""
                },
                {
                    "name": "DELETION_FAILED",
                    "value": "DELETION_FAILED",
                    "documentation": ""
                },
                {
                    "name": "DELETED",

 "value": "DELETED",
                    "documentation": ""
```

```
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
              "name": "NOT_APPLICABLE",
              "value": "NOT_APPLICABLE",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
        "type": "list",
         "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
```

```
            }

    }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
```

```
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
```

```
            "method": "POST",
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
 from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
 <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
```

contain details about the instances that provide user-based subscriptions and also meet the request criteria. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects. "
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " List the Remote Desktop Services (RDS) License Server endpoints ",

```
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoints": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },

        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
```

```
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

  },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
  you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
```

```
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "InstanceUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }

},
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output":
{}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

           },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
      "type": "union",

     "members": {
        "RdsSalSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }

    },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                 "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
```

```
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }

      }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {

   "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
        "smithy.api#http": {
```

```
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
```

```
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
       },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
               "method": "POST",
               "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
               }
            },
            "IdentityProviderArn":
 {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
               }
            },
            "UpdateSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
               "traits": {
                  "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
                  "smithy.api#required": {}
```

```
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
          }
```

```
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
```

```
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
```

```
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan
```

```
on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
```

```
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType
```

```
 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
```

```
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
```

```
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
```

```
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
```

```
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
```

```
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
```

```
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string
```

```
 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

```
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string
```

```
 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
```

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/DeregisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
```

```
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
          "smithy.api#http": {
```

```
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            },
```

```
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
```

```
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {}
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
```

```
data for Linux\n      subscriptions.</p>"
                }
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
            "type": "structure",
            "members": {
              "AmiId": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
                }
              },
              "InstanceID": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
                }
              },
              "InstanceType": {
                "target": "smithy.api#String",
                "traits":
{
                  "smithy.api#documentation": "<p>The instance type of the resource.</p>"
                }
              },
              "AccountID": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
                }
              },
              "Status": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The status of the instance.</p>"
                }
              },
              "Region": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
                }
              },
              "UsageOperation": {
                "target": "smithy.api#String",
```

```
   "traits": {
                    "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
                }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
            },
            "LastUpdatedTime": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
               }
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
```

```
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
         "type": "service",
         "version": "2018-05-10",
         "operations": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
       "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
```

```
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
```

```
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }

        ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]

}
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
```

```
                    },
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
```

```
                "ref": "UseFIPS"
                    },
                    true
                ]
                },
                {
                "fn": "booleanEquals",
                "argv": [
                    {
                    "ref": "UseDualStack"
                    },
                    true
                ]
                }

],
            "rules": [
                {
                "conditions": [
                    {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                        "fn": "getAttr",
                        "argv": [
                            {
                            "ref": "PartitionResult"

},
                            "supportsFIPS"
                        ]
                        }
                    ]
                    },
                    {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                        "fn": "getAttr",
                        "argv": [

                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
```

```
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                    "type": "tree"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {

        "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }
```

```
                ],
                "rules": [
                  {
                    "conditions": [
                      {

                "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
                          "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

  "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
                      ],
                      "type": "tree"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled
but this partition does not support FIPS",
                  "type": "error"
```

```
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true

    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {

                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

{
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
```

```
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                    }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
],
"type": "tree"
```

```
      }
    ]
  },
  "smithy.rules#endpointTests": {
    "testCases": [
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

    }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {

     "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
```

```
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
              }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
```

```
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
```

```
        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {

        "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",

          "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
        "version": "1.0"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type": "operation",
```

```
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n          <ul>\n            <li>\n              <p>\n                <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n              <p>\n                <code>AmiID</code>\n              </p>\n            </li>\n
          <li>\n              <p>\n                <code>DualSubscription</code>\n              </p>\n            </li>\n
```

\n \n <code>InstanceID</code>\n \n \n \n \n <code>InstanceType</code>\n \n \n \n \n <code>ProductCode</code>\n \n \n \n \n <code>Region</code>\n \n \n \n \n <code>Status</code>\n \n \n \n \n <code>UsageOperation</code>\n \n \n \n For each filter, you can use one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": " The maximum items to return in a request. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": " An array that contains instance objects. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield

isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
objects. "
}
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
"type": "operation",
"input": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
},
"errors": [
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
},
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
"smithy.api#documentation":
" Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations. ",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
"inputToken": "NextToken",
"outputToken": "NextToken",
"pageSize": "MaxResults",
"items": "Subscriptions"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
"type": "structure",
"members": {
"Filters": {

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
```

```
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
```

```
the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
```

```
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
          "code": 200,
```

```
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
              "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

              "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
```

\n\t\t\tthis is the status of the offline token. "
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status. "
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": " The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider. "
            }
          }
        },
        "traits": {
          "smithy.api#documentation": " A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider. "
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": " Unable to find the requested Amazon Web Services resource. ",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-

```
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
             },
            {
              "name": "Completed",
              "value": "Completed",
              "documentation": "Completed status"
            },
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
            }
          }
         },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          },
          "smithy.api#uniqueItems": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
```

```
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
{}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
```

```
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
{}
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowUpdate": {
              "target": "smithy.api#Boolean",
              "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
              }
            }

          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
```

```
from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
              "message": {
                  "target": "smithy.api#String"
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
              "smithy.api#error": "client"
          }
      }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.43 protobuf-go 1.36.10

1.43.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.44 github.com/aws/aws-sdk-go-v2/aws/protocol/eventstream 1.7.2

1.44.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing

```
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
```

```
//
// This member is required.
Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string
```

```
 // Date and time range during which the license is valid, in ISO8601-UTC format.
 //
 // This member is required.
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
```

ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd

Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
```

```
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
```

```
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
```

```
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
```

```
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
```

```
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
```

```
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
```

OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
```

```
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
            }
```

```
            },
            "DomainNetworkSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "SELF_MANAGED",
                   "value": "SELF_MANAGED",
                  "documentation": "SELF_MANAGED type of Active Directory"
               },
               {
                  "name": "AWS_MANAGED",
                  "value": "AWS_MANAGED",
                  "documentation": "AWS_MANAGED type of Active Directory"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
         },
         "output": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
         },
         "errors": [
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                  "smithy.api#required": {}
               }
```

```
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
               }
            },
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
          "smithy.api#error": "server"
        }

    },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/CreateLicenseServerEndpoint"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
 {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
              "smithy.api#required": {}
            }
          },
          "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
      },
      "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}

    }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

     },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/DeregisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
            "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
            },
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Value of the filter.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
            "ActiveDirectoryIdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
                }
```

```
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

                 "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }
```

```
            },
            "FailureMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
               }
            },
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                  "smithy.api#required": {}
               }
            },
            "Products": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>A list of provided user-based subscription
```

products. ",
                "smithy.api#required": {}
              }
            },
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": " The date of the last status check. "
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": " The status message for an EC2 instance. "
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": " The AWS Account ID of the owner of this resource. "
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                "smithy.api#documentation": " The <code>IdentityProvider</code> resource specifies details about the identity provider. "
              }
            }
          },
          "traits": {
            "smithy.api#documentation": " Describes an EC2 instance providing user-based subscriptions. "
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": " The user name

```
 from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
                }
            },
```

```
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
```

```
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
```

```
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
```

```
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
```

```
                },

            "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "aws.partition",
                        "argv": [
                            {
                                "ref": "Region"
                            }
                        ],
                        "assign": "PartitionResult"
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },

true
                                ]
                            },
                            {
```

```
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

"fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      }
                    ]
                  },

                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }

              ]
                  }
                ],
                "rules": [
                  {
```

```
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

            ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [

            {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "fn": "getAttr",
                                                "argv": [

                                {
                                                        "ref": "PartitionResult"
                                                    },
```

```
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {

    "conditions": [
                        {
                            "fn": "booleanEquals",
```

```
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [
                      {
                        "ref": "PartitionResult"
                      },
                      "supportsDualStack"
                    ]

            }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }

            ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "DualStack is enabled but this partition does not support DualStack",
            "type": "error"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
          "headers": {}
        },
        "type": "endpoint"
```

```
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",

        "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
```

```
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS":
 true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
```

```
"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint":
{
                        "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }
                },
                "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

      "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
```

```
                "url": "https://example.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,

      "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "UseFIPS": false,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
```

```
                    }
                },
                {
                    "documentation": "Missing region",
                    "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
```

```
server.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
            "type": "structure",
            "members": {
                "IdentityProviderArn": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                    }
                },
                "ServerType": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                    "traits": {
                        "smithy.api#documentation": "<p>The type of license server.</p>"
                    }
                },
                "ServerEndpoint": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                    "traits": {
                        "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                    }
                },
                "StatusMessage": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                    }
                },
                "LicenseServerEndpointId": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                    "traits": {
                        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                    }
                },
                "LicenseServerEndpointArn": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                    "traits": {
                        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                    }
                },
                "LicenseServerEndpointProvisioningStatus": {
                    "target":
```

```
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
```

```
                "value": "PROVISIONING_FAILED",
                "documentation": ""
              },
              {
                "name": "PROVISIONED",
                "value": "PROVISIONED",
                "documentation": ""
              },
              {
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
       "member": {
         "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
          }

  }
      },
      "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
              }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources that contain details about the Active Directory identity providers that meet the request criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
             }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
```

from a previously truncated response. "
}
},
"Filters": {
"target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
"traits": {
"smithy.api#documentation": " You can use the following filters to streamline results: <ul> Status   InstanceId "
}
}
},
"traits": {
"smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
"type": "structure",

"members": {
"InstanceSummaries": {
"target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
"traits": {
"smithy.api#documentation": " An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects. "
}
}
},
"traits": {
"smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
"type": "operation",
"input": {
"target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
},
"errors": [

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
          "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
```

```
                }
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
          "type": "structure",
          "members": {
            "LicenseServerEndpoints": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
              "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits":
 {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
```

```
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
```

```
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

},
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
        "smithy.api#http": {
          "method": "GET",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#readonly": {}
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}

 }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
```

```
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
```

```
        "traits": {
          "smithy.api#output":
 {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

   "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
```

```
product for this user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
```

```
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }

         },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
```

```
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
```

```
        "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
          },
          "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
          "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

       "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
```

```
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Settings": {
          "type": "structure",
          "members": {
            "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }

      },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
          "type": "operation",
```

```
        "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
```

```
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
           "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
       },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
         },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
```

```
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
```

```
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {

     "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
```

```
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
```

```
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
```

```
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
        "smithy.api#http": {

  "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "IdentityProviderArn":
 {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
              }
            },
            "UpdateSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
              "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                "smithy.api#required": {}
              }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
```

```
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
```

```
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string
```

```
 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
 return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
 return err
 }
 if err = addRetry(stack, options); err != nil {
 return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
```

```
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
```

```
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
```

```
 params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string
```

```
 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
```

```
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
```

```
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
```

```
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
```

```
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType
```

```
 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
```

```
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
```

```
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
```

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
           "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
           },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
```

```
        "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
```

```
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
 {
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
```

```
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
```

```
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }

        ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
```

```
                {
                  "conditions": [
                    {

              "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
                      "assign": "PartitionResult"
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {

                        "ref": "UseFIPS"
                            },
                            true
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }

],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
```

```
                                        "ref": "PartitionResult"

                    },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [

                                     {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {

      "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

            "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled
but this partition does not support FIPS",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true

        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
```

```
                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

        {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
  }
 ],
 "type": "tree"
 }
 ]
},
"smithy.rules#endpointTests": {
 "testCases": [
  {
   "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
   "expect": {
    "endpoint": {
     "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

 }
   },
   "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
   }
  },
  {
   "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
   "expect": {
    "endpoint": {
     "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
   },
   "params": {
    "Region": "us-east-1",
```

```
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                }

    },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
```

```
"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                      }
                     },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
              "Region": "us-isob-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "Missing region",

                "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
```

```
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
```

```
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
```

" Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations. ",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
"inputToken": "NextToken",
"outputToken": "NextToken",
"pageSize": "MaxResults",
"items": "Subscriptions"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
"type": "structure",
"members": {
"Filters": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

"traits": {
"smithy.api#documentation": " An array of structures that you can use to filter the results to those that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n <code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't match a certain subscription's name. \n The valid names for this filter are: \n \n \n \n <code>Subscription</code>\n \n \n \n The valid Operators for this filter are: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
}
},
"MaxResults": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
"traits": {
"smithy.api#documentation": " The maximum items to return in a request. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
"smithy.api#length": {
"min": 1,
"max": 16384

```
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
                  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
 return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
 nextToken from a previously truncated response.</p>"
              }
            }
          },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
```

```
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
        "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
                }
            },
            "SubscriptionProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
```

```
        "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
                "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
        },
        "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
            }
        },
        "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
        },
        "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
        },
        "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

                "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
               },
              {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
              },
              {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
              },
              {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
              }
            ]
```

```
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
                }
```

```
            },
            "InstanceCount": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
               "traits": {
                  "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
         "type": "list",

       "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "RedHat",
                  "value": "RedHat",
                  "documentation": "RedHat subscription provider namespace"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
         "type":
 "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
```

```
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
 {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
```

```
Region.</p>"
                }
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
```

```
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.45 github.com/aws/aws-sdk-go-v2/service/internal/accept-encoding 1.13.2

1.45.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
```

```
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
```

```
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
```

```
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
```

```
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent

AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

```
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {
```

```
 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
```

```
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
```

```
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {
```

```
 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
```

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
        "type": "structure",
        "members": {
          "DirectoryId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
            "traits": {
              "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
          },
          "ActiveDirectorySettings":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

           "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
```

```
        "members": {
            "DomainName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
                }
            },
            "DomainIpv4List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainIpv6List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
             "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
          }

 },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

         {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
{
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                    "smithy.api#required": {}

  }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
```

```
        "LicenseServerEndpoint": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
          "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

     },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
```

```
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
```

```
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
              }
            }
          },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },

    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
```

```
            },
            "Operation": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
               }
            },
            "Value": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Value of the filter.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
         "type": "union",
         "members": {
            "ActiveDirectoryIdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
```

```
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
          }
        }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
```

```
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {

                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
```

```
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
         },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
         },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
        },
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
```

```
    "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The AWS region used to dispatch the request.",
                    "type": "string"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                    "type": "boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                    "type": "boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "string"
                }
            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]

    }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
```

```
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },

            "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
```

```
                {
                  "ref": "Region"
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
```

```
        "fn": "booleanEquals",
                                    "argv": [
                                      true,
                                      {
                                        "fn": "getAttr",
                                        "argv": [
                                          {
                                            "ref": "PartitionResult"
                                          },
                                          "supportsFIPS"
                                        ]
                                      }
                                    ]
                                  },

                                  {
                                    "fn": "booleanEquals",
                                    "argv": [
                                      true,
                                      {
                                        "fn": "getAttr",
                                        "argv": [
                                          {
                                            "ref": "PartitionResult"
                                          },
                                          "supportsDualStack"
                                        ]
                                      }

                                ]
                                    }
                                  ],
                                  "rules": [
                                    {
                                      "conditions": [],
                                      "endpoint": {
                                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                      },
                                      "type": "endpoint"
                                    }

        ],
                                  "type": "tree"
                                },
                                {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [

        {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [

              {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {

                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                      "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"

}
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

        }
                            ]
                        }
```

```
                                    ],
                                    "rules": [
                                        {
                                            "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }

                                    ],
                                    "type": "tree"
                                },
                                {
                                    "conditions": [],
                                    "error": "DualStack is enabled but this partition does not support DualStack",
                                    "type": "error"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
```

```
        "smithy.rules#endpointTests": {
          "testCases": [
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",

          "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
```

```
    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
```

```
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
```

```
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,

         "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
```

```
disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
                      }
                    },
                    "params": {
                      "UseFIPS": false,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                    "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                     }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
      "type": "structure",
      "members": {
        "ProvisioningStatus": {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
          "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
          }
        },
        "HealthStatus": {
          "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
          "traits": {
            "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
          }
        },
        "Ipv4Address": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
          }
        },
        "Ipv6Address": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
```

```
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>"
            }
        },
        "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
            }
        },
        "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
        },
        "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
        },
        "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
        },
```

```
            "CreationTime": {
               "target": "smithy.api#Timestamp",

      "traits": {
                  "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
         "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {

               "name": "PROVISIONING",
                  "value": "PROVISIONING",
                  "documentation": ""
               },
               {
                  "name": "PROVISIONING_FAILED",
                  "value": "PROVISIONING_FAILED",
                  "documentation": ""
               },
               {
                  "name": "PROVISIONED",
                  "value": "PROVISIONED",
                  "documentation": ""
               },
               {
                  "name": "DELETING",
                  "value": "DELETING",
                  "documentation": ""
               },
```

```
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
              "name": "DELETED",

"value": "DELETED",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
              "name": "NOT_APPLICABLE",
              "value": "NOT_APPLICABLE",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
        "type": "list",
         "member": {
           "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }

  }
      },
      "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
         {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
             "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
             }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
```

```
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
```

```
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits":
{
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
```

```
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
```

```
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

}
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product
```

```
user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
                }
            },
            "SubscriptionStartDate": {
                "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
 reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
 license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
```

```
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

          },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"

        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
      "type": "structure",
      "members": {
        "SecretId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
            "smithy.api#length": {
              "min": 1
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 5,
          "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
      "type": "structure",
      "members": {
        "Endpoint": {
```

```
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Settings": {
          "type": "structure",
          "members": {
            "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }

        },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
          "type": "operation",
          "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
```

```
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
```

```
            },
            "IdentityProvider":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
                 }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                    "smithy.api#required": {}
                }

     }
        },
        "traits": {
```

```
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {

    "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
```

```
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
          "smithy.api#http": {

    "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
            }
```

```
        },
        "UpdateSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
          }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
              }
            }
          },

          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
          }
        }
      }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
```

```
 params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
        region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
```

```
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
```

```
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData
```

```
// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
```

```
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}
```

```
type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string
```

```
 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
```

```
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
```

```
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
```

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

```
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {
```

```
 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
```

```
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
```

```
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },

        {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                 "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
```

```
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
```

```
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
```

```
            "Region": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
               }
            },
            "UsageOperation": {
               "target": "smithy.api#String",

   "traits": {
                  "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
               }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
            },
            "LastUpdatedTime": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
               }
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
```

```
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
                }
            },
            "SubscriptionProviderUpdateTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
                }
            },
            "DualSubscription": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
                }
            },
            "RegisteredWithSubscriptionProvider": {
                "target": "smithy.api#String",
                "traits": {

      "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
       },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
         "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
```

```
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
```

```
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }

],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
                                        },
                                        true
                                    ]
```

```
}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

          }
          ],
          "type": "tree"
        },
        {
          "conditions": [],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Region"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

          "fn": "aws.partition",
                        "argv": [
                          {
                            "ref": "Region"
                          }
                        ],
```

```
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {

                    "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }

],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"

            },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
```

```
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [

                     {
                          "ref": "PartitionResult"
                      },
                      "supportsDualStack"
                    ]
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],

    "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],

          "type": "tree"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
              "type": "error"
            }
          ],
          "type": "tree"
        },
        {
```

```
            "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

          "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
                          "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                            "headers": {}
                          },
                          "type": "endpoint"
                        }
                      ],
```

```
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled
but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true

            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {

                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "rules": [
```

```
                                        {
                                            "conditions": [],
                                            "rules": [

        {
                                                    "conditions": [],
                                                    "endpoint": {
                                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                        "properties": {},
                                                        "headers": {}
                                                    },
                                                    "type": "endpoint"
                                                }
                                            ],
                                            "type": "tree"

                                }
                                    ],
                                    "type": "tree"
                                },
                                {
                                    "conditions": [],
                                    "error": "DualStack is enabled but this partition does not support DualStack",
                                    "type": "error"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"
                        }
                    ],
                    "type": "tree"
```

```
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ],
        "type": "tree"
    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

  }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
```

```
                "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
              }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
           },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
```

```
            }

        },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
```

```
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
```

```
                },
                {
                  "documentation": "Missing region",

                "expect": {
                      "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
   <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
   <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
          <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
```

```
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
```

```
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
```

```
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
```

```
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

     "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
              }
            }
          },

          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
          "type": "structure",
          "members": {
            "RegisteredSubscriptionProviders": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
```

```
      "type": "structure",
      "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "tags": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "EQUAL",
            "value": "Equal",
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
```

```
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {

"name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

      {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {

             "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
          }
        },
        "SubscriptionProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
 subscription provider \n\t\t\tthat you requested.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
 Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {

                "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
             },
            {
              "name": "Completed",
              "value": "Completed",
              "documentation": "Completed status"
            },
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
```

```
            }
          }
        },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          },
          "smithy.api#uniqueItems": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with
Amazon EC2, Bring Your Own\n        Subscription model (BYOS), or from the Amazon Web Services Marketplace.
Certain subscriptions may use licensing from the\n        Amazon Web Services Marketplace as well as OS licensing
from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

       "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
```

```
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
```

```
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",
```

```
    "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tagKeys": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input":
 {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
```

```
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
{}
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowUpdate": {
              "target": "smithy.api#Boolean",
              "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
              }
            }

          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
          "type": "structure",
          "members": {
```

```
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
                }
             }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
          "smithy.api#error": "client"
```

```
        }
      }
    }
  }
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.46 iputils-clockdiff 20240905-r0

1.46.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest

your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any

such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to
decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing
version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU
General Public License as published by
the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc.,

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.47 github.com/aws/aws-sdk-go-v2/config 1.31.15

1.47.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger

OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}
```

```
type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
 return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
 return err
 }
 if err = addRetry(stack, options); err != nil {
 return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
```

```
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware

```
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
```

```
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
```

```
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
```

```
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
```

```
//   - ProductSKU
//
//   - Fingerprint
//
//   - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
```

```
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
          "type": "structure",
          "members": {
            "DirectoryId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
              "traits": {
                "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
              }
            },
            "ActiveDirectorySettings":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
              }
            },
            "ActiveDirectoryType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
              "traits": {
                "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
              }
            },
            "IsSharedActiveDirectory": {
              "target": "smithy.api#Boolean",
              "traits": {

           "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
          "type": "structure",
          "members": {
            "DomainName": {
              "target": "smithy.api#String",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
        },
        "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
        },
        "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
```

```
            "smithy.api#enum": [
               {
                  "name": "SELF_MANAGED",
                   "value": "SELF_MANAGED",
                  "documentation": "SELF_MANAGED type of Active Directory"
               },
               {
                  "name": "AWS_MANAGED",
                  "value": "AWS_MANAGED",
                  "documentation": "AWS_MANAGED type of Active Directory"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
         },
         "output": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
 {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
               }
            },
```

```
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                 "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
            "smithy.api#error": "server"
        }
```

```
},
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

      {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/CreateLicenseServerEndpoint"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
```

```
{
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                  "smithy.api#required": {}
               }
            },
            "LicenseServerSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
               }
            },
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
               }
            }
```

```
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/DeleteLicenseServerEndpoint"

          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "LicenseServerEndpoint": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
              "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
```

```
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

    },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
```

```
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },

    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
              }
            },
            "Value": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
                "smithy.api#required": {}
              }
```

```
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
            "smithy.api#required": {}
          }
        },
        "Products": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
            "smithy.api#required": {}
          }
        },
        "LastStatusCheckDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                "smithy.api#required": {}
            }
```

```
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {

                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
```

```
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
```

```
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
 },
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
 },
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
```

```
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
```

```
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
```

```
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

{
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },

    "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
```

```
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "aws.partition",
                        "argv": [
                            {
                                "ref": "Region"
                            }
                        ],
                        "assign": "PartitionResult"
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },

true
                                ]
                            },
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {

"fn": "booleanEquals",
                                        "argv": [
                                            true,
                                            {
                                                "fn": "getAttr",
```

```
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            },

            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }

            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
```

```
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [

{
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                    {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
```

```
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"

}
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

    "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

     }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
```

```
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                    "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
```

```
"expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",

              "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

   "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
```

```
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
```

```
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,

         "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
```

```
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server.</p>"
            }
        },
        "HealthStatus": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>"
            }
        },
```

```
        "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
            }
        },
        "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
        },
        "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
        },
        "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
        },
        "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
```

```
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }

  }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
```

```
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
             "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
             }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
```

```
criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
             }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",
```

```
      "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
                }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
         "type": "structure",
         "members": {
            "ProductUserSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
```

```
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
        },
        "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
        },
        "SubscriptionEndDate": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
 {
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }
```

```
        },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
```

```
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
      "type": "structure",
      "members": {
        "SecretId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
            "smithy.api#length": {
              "min": 1
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 5,
          "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
      "type": "structure",
      "members": {
        "Endpoint": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
```

```
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
 {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
```

```
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }

         }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StringList": {
          "type": "list",
```

```
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

"min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
```

```
        ],
        "traits": {
          "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
```

```
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

            }
            },
            "TagKeys": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
               "method": "POST",
               "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
               }
            },
            "IdentityProviderArn":
 {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
               }
            },
            "UpdateSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
               "traits": {
                  "smithy.api#documentation": "<p>Updates the registered identity providers product related
```

configuration settings. You can update any combination of settings in a single operation such as the: <ul> Subnets which you want to add to provision VPC endpoints.   Subnets which you want to remove the VPC endpoints from.   Security group ID which permits traffic to the VPC endpoints. ",
"smithy.api#required": {}
}
}
},
"traits": {
"smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
"type": "structure",
"members": {
"IdentityProviderSummary": {
"target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
"traits": {
"smithy.api#required": {}
}
}
},
"traits": {
"smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
"type": "structure",
"members": {
"AddSubnets": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
"traits": {
"smithy.api#documentation": " The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers. ",
"smithy.api#required": {}
}
},
"RemoveSubnets": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
"traits": {
"smithy.api#documentation": " The ID of one or more subnets to remove. ",
"smithy.api#required": {}
}
},
"SecurityGroupId": {
"target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",

```
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
```

```
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
```

```
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region:
      region,
 ServiceID:     ServiceID,
 OperationName: "GetLicense",
```

```
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
```

```
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
```

```
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string
```

```
// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata
```

```
// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
```

```
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options)) (*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
```

```
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
```

```
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
```

```
//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string
```

```
 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
         "type": "structure",
         "members": {
            "SubscriptionProviderArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
         "type": "structure",
         "members": {},
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
         "type": "structure",
         "members": {
            "Name": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The type of name to filter by.</p>"
               }
            },
```

```
            "Values": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
                }
            },
            "Operator": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
                "traits": {
                    "smithy.api#documentation": "<p>An operator for filtering results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
 "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
             },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
```

```
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
           "traits": {
             "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
```

```
            },
            "UsageOperation": {
               "target": "smithy.api#String",

   "traits": {
                  "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
               }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
            },
            "LastUpdatedTime": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
               }
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
          },
          "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
```

```
"operations": [
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
         "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
```

```
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
```

```
                {
                  "conditions": [
                    {
                      "fn": "isSet",
                      "argv": [
                        {
                          "ref": "Endpoint"
                        }
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseFIPS"
                            },
                            true
                          ]
                        }

        ],
                      "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                  "ref": "UseDualStack"
                                },
                                true
                              ]

}
                          ],
                          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                          "type": "error"
                        },
                        {
```

```
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

        "fn": "aws.partition",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ],
                    "assign": "PartitionResult"
                  }
                ],
                "rules": [
                  {
                    "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {

                "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }

    ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"

            },
                                "supportsFIPS"
                            ]
                          }
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
```

```
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {

        "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
```

```
                },
                true
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {

        "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true

                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                }
              ],
              "type": "tree"
            },
            {
```

```
                "conditions": [],
                "error": "FIPS is enabled
but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true

        ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {

                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [

{
                        "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
```

```
                "type": "error"
            }
          ],
          "type": "tree"
        }
      ]
    },
    "smithy.rules#endpointTests": {
      "testCases": [
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

      }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

       "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
```

```
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
          }

    },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {

  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
```

```
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }

       },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
```

```
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
```

```
      }
    },

  {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

   "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
```

```
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
 <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
```

```
streamline results:</p>\n <ul>\n <li>\n <p>\n <code>AccountID</code>\n
</p>\n </li>\n <li>\n <p>\n <code>AmiID</code>\n </p>\n </li>\n
<li>\n <p>\n <code>DualSubscription</code>\n </p>\n </li>\n
<li>\n <p>\n <code>InstanceID</code>\n </p>\n </li>\n <li>\n
<p>\n <code>InstanceType</code>\n </p>\n </li>\n <li>\n <p>\n
<code>ProductCode</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Region</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Status</code>\n </p>\n </li>\n <li>\n <p>\n
<code>UsageOperation</code>\n </p>\n </li>\n </ul>\n <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n <ul>\n
<li>\n <p>\n <code>contains</code>\n </p>\n </li>\n <li>\n
<p>\n <code>equals</code>\n </p>\n </li>\n <li>\n
<p>\n <code>Notequal</code>\n </p>\n </li>\n </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 16384
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
```

```
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n       <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n        <p>The valid names for this filter are:</p>\n        <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n        </ul>\n        <p>The valid Operators for this filter are:</p>\n        <ul>\n          <li>\n            <p>\n        <code>contains</code>\n          </p>\n        </li>\n        <li>\n          <p>\n        <code>equals</code>\n          </p>\n        </li>\n        <li>\n          <p>\n        <code>Notequal</code>\n          </p>\n        </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"

               }
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
         "type": "operation",
         "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
         },
         "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "RegisteredSubscriptionProviders"
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
```

```
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "tags": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "EQUAL",
            "value": "Equal",
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {

"name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },

     {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
```

```
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
               "traits": {
                 "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                 "smithy.api#required": {}
               }
            },
            "SecretArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
              "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've
stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat
Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                 "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                 "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
```

```
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
             },
            {
              "name": "Completed",
              "value": "Completed",
              "documentation": "Completed status"
            },
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
            }
          }
         },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
```

```
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
    {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
    resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
```

```
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
```

```
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
```

```
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
{}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
```

```
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
                }
             }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.48 iputils-arping 20240905-r0

1.48.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which

gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the
Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot

impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing
version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU
General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.49 apk-tools 2.14.9-r3

1.49.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to

most of the Free Software Foundation's software and to any other program whose
authors commit to using it. (Some other Free Software Foundation software
is covered by the GNU Lesser General Public License instead.) You can apply
it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our
General Public Licenses are designed to make sure that you have the
freedom
to distribute copies of free software (and charge for this service if you
wish), that you receive source code or can get it if you want it, that you
can change the software or use pieces of it in new free programs; and that
you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to
deny you these rights or to ask you to surrender the rights. These restrictions
translate to certain responsibilities for you if you distribute copies of
the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or
for a fee, you must give the recipients all the rights that you have. You
must make sure that they, too, receive or can get the source code. And you
must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2)
offer you this license which gives you legal permission to copy, distribute
and/or modify the software.

Also, for
each author's protection and ours, we want to make certain that
everyone understands that there is no warranty for this free software. If
the software is modified by someone else and passed on, we want its recipients
to know that what they have is not the original, so that any problems introduced
by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We
wish to avoid the danger that redistributors of a free program will individually
obtain patent licenses, in effect making the program proprietary. To prevent
this, we have made it clear that any patent must be licensed for everyone's
free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification
follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice
placed by the copyright holder saying it may be distributed under

the terms
of this General Public License. The "Program", below, refers to any such program
or work, and a "work based on the Program" means either the Program or any
derivative work under copyright law: that is to say, a work containing the
Program or a portion of it, either verbatim or with modifications and/or translated
into another language. (Hereinafter, translation is included without limitation
in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered
by this License; they are outside its scope. The act of running the Program
is not restricted, and the output from the Program is covered only if its
contents constitute a work based on the Program (independent of having been
made by running the Program). Whether that is true depends on what the Program
does.

1. You may copy and distribute verbatim copies of the Program's source code
as you receive it, in any medium, provided that you conspicuously
and appropriately
publish on each copy an appropriate copyright notice and disclaimer of warranty;
keep intact all the notices that refer to this License and to the absence
of any warranty; and give any other recipients of the Program a copy of this
License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you
may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it,
thus forming a work based on the Program, and copy and distribute such modifications
or work under the terms of Section 1 above, provided that you also meet all
of these conditions:

a) You must cause the modified files to carry prominent notices stating that
you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or
in part contains or is derived from the Program or any part thereof, to be
licensed as a whole at no charge
to all third parties under the terms of this
License.

c) If the modified program normally reads commands interactively when run,
you must cause it, when started running for such interactive use in the most
ordinary way, to print or display an announcement including an appropriate
copyright notice and a notice that there is no warranty (or else, saying that
you provide a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this License.
(Exception: if the Program itself is interactive but does not normally print
such an announcement, your work based on the Program is not required to print

an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.
But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it
with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means
all
the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation

of the executable. However, as a special exception, the source code distributed
need not include anything that is normally distributed (in either source or
binary form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component itself
accompanies the executable.

If distribution of executable or object code is made by offering access to
copy from a designated place, then offering equivalent access to copy the
source code from the same place counts as distribution of the source code,
even though third parties are not compelled to copy the source along with
the object code.

4. You may not copy, modify, sublicense, or distribute the Program except
as expressly provided under this License. Any attempt otherwise to copy, modify,
sublicense or distribute the Program
is void, and will automatically terminate
your rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses terminated
so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed
it. However, nothing else grants you permission to modify or distribute the
Program or its derivative works. These actions are prohibited by law if you
do not accept this License. Therefore, by modifying or distributing the Program
(or any work based on the Program), you indicate your acceptance of this License
to do so, and all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program),
the recipient automatically receives a license from the original licensor
to copy, distribute or modify the Program subject to these terms and conditions.
You may
not impose any further restrictions on the recipients' exercise of
the rights granted herein. You are not responsible for enforcing compliance
by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement
or for any other reason (not limited to patent issues), conditions are imposed
on you (whether by court order, agreement or otherwise) that contradict the
conditions of this License, they do not excuse you from the conditions of
this License. If you cannot distribute so as to satisfy simultaneously your
obligations under this License and any other pertinent obligations, then as
a consequence you may not distribute the Program at all. For example, if a
patent license would not permit royalty-free redistribution of the Program
by all those who receive copies directly or indirectly through you, then the
only way you could satisfy both it and this License would be to refrain entirely
from distribution of the Program.

If any portion of
this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply and
the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents
or other property right claims or to contest validity of any such claims;
this section has the sole purpose of protecting the integrity of the free
software distribution system, which is implemented by public license practices.
Many people have made generous contributions to the wide range of software
distributed through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing to
distribute software through any other system and a licensee cannot impose
that choice.

This section is intended to make thoroughly clear what is believed to be a
consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted
in certain
countries either by patents or by copyrighted interfaces, the original copyright
holder who places the Program under this License may add an explicit geographical
distribution limitation excluding those countries, so that distribution is
permitted only in or among countries not thus excluded. In such case, this
License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of
the General Public License from time to time. Such new versions will be similar
in spirit to the present version, but may differ in detail to address new
problems or concerns.

Each version is given a distinguishing version number. If the Program specifies
a version number of this License which applies to it and "any later version",
you have the option of following the terms and conditions either of that version
or of any later version published by the Free Software Foundation. If the
Program does not specify a version
number of this License, you may choose
any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs
whose distribution conditions are different, write to the author to ask for
permission. For software which is copyrighted by the Free Software Foundation,
write to the Free Software Foundation; we sometimes make exceptions for this.
Our decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing and reuse
of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best
way to achieve this is to make it free software
which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a

copy of the GNU General Public License along with
this program; if not, write to the Free Software Foundation, Inc., 51 Franklin
Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when
it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes
with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software,
and you are welcome to redistribute it under certain conditions; type `show
c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may be
called something other than `show w' and `show c'; they could even be mouse-clicks
or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school,
if any, to sign a "copyright
disclaimer" for the program, if necessary. Here
is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision'
(which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General
Public License does not permit incorporating your program into proprietary
programs. If your program is a subroutine library, you may consider it more
useful to permit linking proprietary applications with the library. If this
is what you want to do, use the GNU Lesser General Public License instead
of this License.

1.50 kelseyhightower-envconfig 1.4.0

1.50.1 Available under license :

Copyright (c) 2013 Kelsey Hightower

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.51 go.yaml.in/yaml/v2 2.4.3

1.51.1 Available under license :

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,

indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.52 aws-sdk-go-v2-service-sso 1.29.8

1.52.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

```
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
```

```
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
```

```
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
```

```
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart

```
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string
```

```
 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
```

```
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
```

```
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
```

```
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }
```

```
 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
```

```
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
```

```
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
            "DomainName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
                }
            },
            "DomainIpv4List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainIpv6List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
```

```
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
 {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
               "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
```

```
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
          "smithy.api#error": "server"
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
```

```
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
              }
            },
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
          "type": "union",
          "members": {
            "SecretsManagerCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
```

```
            {

          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/license-server/DeleteLicenseServerEndpoint"

           },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
          "type": "structure",
          "members": {
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}

    }
            }
          },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoint": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
          "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

     },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
```

```
 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
```

```
  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
               }
            },
            "InstanceUserArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
```

```
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

                 "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }

            },
            "FailureMessage": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
        },
        "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        },
        "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                "smithy.api#required": {}
            }
        },
```

```
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
          },
          "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
```

```
            },
            "DisassociationDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
         "type": "list",
         "member": {

     "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
        },
        {

            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
    ],
    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager User Subscriptions",
            "arnNamespace": "license-manager-user-subscriptions",
            "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                 "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
```

```
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
```

```
                "ref": "Endpoint"
              }
            ]

  }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
          },
          {
            "conditions": [

{
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
          },
          {
            "conditions": [],
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },

      "properties": {},
                "headers": {}
              },
```

```
                "type": "endpoint"
            }
          ],
          "type": "tree"
        },
        {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Region"
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
```

```
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

"fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    },

                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
```

```
                "headers": {}
              },
              "type": "endpoint"
            }

        ],
          "type": "tree"
        },
        {
          "conditions": [],
          "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
          "type": "error"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [

    {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "fn": "getAttr",
                  "argv": [

        {
                      "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                  ]
                },
                true
              ]
```

```
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {

    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
```

```
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]

            }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    }
],
```

```
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",

            "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

    "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
 true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
```

```
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
```

```
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                  }
                },
                "params": {
                  "Region": "us-iso-east-1",
                  "UseFIPS":
false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

      "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
```

```
                "UseFIPS": false,

        "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
               }
            },
            {
              "documentation": "Missing region",
              "expect": {
```

```
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
     }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
```

```
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
```

```
        },
        "LicenseServers": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
          }
        },
        "CreationTime": {
          "target": "smithy.api#Timestamp",

    "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
```

```
                "value": "PROVISIONED",
                "documentation": ""
              },
              {
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

      {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
```

```
operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

},
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
```

```
            "ResourceArn": {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                  "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
         "type": "structure",
         "members": {
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}

 }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
```

```
    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
 {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS) license server.</p>"
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
```

```
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

          },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                  "smithy.api#required": {}
                }
              }
          },
          "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
              "smithy.api#length": {
                "min": 1
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
 {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
```

```
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
```

```
{
        "type": "structure",
        "members": {
           "message": {
              "target": "smithy.api#String"
           }
        },
        "traits": {
           "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
           "smithy.api#error": "client"
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
           "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                 "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                 "smithy.api#length": {
                    "min": 1
                 },
                 "smithy.api#required": {}
              }

     },
           "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                 "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                 "smithy.api#required": {}
              }
           }
        },
        "traits": {
           "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
           "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
```

```
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        },
        "IdentityProvider":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
           }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }

        }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {

    "min": 0,
            "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

            }
            },
            "TagKeys": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
        "smithy.api#http": {

  "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
```

```
            },
            "IdentityProviderArn":
 {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }

            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
```

```
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
```

```
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
```

```
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string
```

```
 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
```

```
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}
```

```
type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
```

```
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string
```

```
 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
```

```
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
```

```
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
```

```
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
           "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
```

```
        "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
```

```
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
 {
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
```

```
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
```

```
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }

        ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
```

```
                {
                  "conditions": [
                    {

            "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
                      "assign": "PartitionResult"
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {

                        "ref": "UseFIPS"
                            },
                            true
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }

],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
```

```
                                        "ref": "PartitionResult"

                    },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [

                                     {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {

      "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

            "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled
but this partition does not support FIPS",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
```

```
                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

        {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
"testCases": [
{
"documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
"expect": {
"endpoint": {
"url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

}
},
"params": {
"Region": "us-east-1",
"UseFIPS": true,
"UseDualStack": true
}
},
{
"documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
"expect": {
"endpoint": {
"url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
}
},
"params": {
"Region": "us-east-1",
```

```
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
                }

    },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
```

```
"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                      }
                     },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
              "Region": "us-isob-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "Missing region",

                  "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
            "type": "string",
            "traits": {
              "smithy.api#enum": [
                {
                  "name": "Enabled",
                  "value": "Enabled",
                  "documentation": "Enabled LinuxSubscriptionsDiscovery"
                },
                {
                  "name": "Disabled",
                  "value": "Disabled",
                  "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
              ]
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
            "type": "structure",
            "members": {
              "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                  "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
```

```
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
```

```
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
```

```
"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                    "min": 1,
                    "max": 16384
```

```
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
                  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
 return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
 nextToken from a previously truncated response.</p>"
              }
            }
          },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
```

```
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
          "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
                }
            },
            "SubscriptionProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
```

```
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
               },
              {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
              },
              {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
              },
              {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
              }
            ]
```

```
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
                }
```

```
            },
            "InstanceCount": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
               "traits": {
                  "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
         "type": "list",

       "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "RedHat",
                  "value": "RedHat",
                  "documentation": "RedHat subscription provider namespace"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
         "type":
 "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
```

```
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
 {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
```

```
Region.</p>"
                }
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
```

```
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.53 openssl 3.5.4-r0

1.53.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under
the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

a) the GNU General Public License as published by the Free
  Software Foundation; either version 1, or (at your option) any
  later version, or
b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

 The GNU General Public License, Version 1, February 1989

                GNU GENERAL PUBLIC LICENSE
                 Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

                     Preamble

 The license agreements of most software companies try to keep users
at the mercy of those companies. By contrast, our General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. The
General Public License applies to the Free Software Foundation's
software and to any other program whose authors commit to using it.
You can use it for your programs, too.

 When we speak of free software, we are referring to freedom, not
price. Specifically, the General Public License is designed to make
sure that you have the freedom to give away or sell copies of free
software, that you receive source code or can get it if you want it,
that you can change the software or use pieces of it in new free
programs; and that you know you can do these things.

 To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender
 the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

 For example, if you distribute copies of a such a program, whether

gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

The precise terms and conditions
for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which
contains a notice placed by the copyright holder saying it may be
distributed under the terms of this General Public License. The
"Program", below, refers to any such program or work, and a "work based
on the Program" means either the Program or any work containing the
Program or a portion of it, either verbatim or with modifications. Each
licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source
code as you receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice and
disclaimer of warranty; keep intact all the notices that refer to this
General Public License and to the absence of any warranty; and give any
other recipients of the
Program a copy of this General Public License
along with the Program. You may charge a fee for the physical act of
transferring a copy.

2. You may modify your copy or copies of the Program or any portion of
it, and copy and distribute such modifications under the terms of Paragraph
1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that
you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that

in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard

libraries that
accompany the operating system on which the executable
file runs, or for standard header files or definitions files that
accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the
Program except as expressly provided under this General Public License.
Any attempt otherwise to copy, modify, sublicense, distribute or transfer
the Program is void, and will automatically terminate your rights to use
the Program under this License. However, parties who have received
copies, or rights to use copies, from you under this General Public
License will not have their licenses terminated so long as such parties
remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based
on the Program) you indicate your acceptance of this license to do so,
and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original
licensor to copy, distribute or modify the Program subject to these
terms and conditions. You may not impose any further restrictions on the
recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of the license which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
the license, you may choose any version ever published by the Free Software
Foundation.

8. If you wish to incorporate parts of the Program into other free
programs whose distribution
conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free
Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the
commands you use may be called something other than `show w' and `show
c'; they could even be mouse-clicks or menu items--whatever suits your
program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here a sample; alter
the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection
of files created through
textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of
the source form of the
Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified
Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names,
and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not
be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.54 aws-sdk-go-v2-aws-protocol-eventstream 1.7.2

1.54.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata

```
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string
```

```
// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
```

```
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
```

```
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput

ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```
 spanInitializeEnd
Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32
```

```
 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
```

```
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
```

```
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

```
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
```

```
 return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
```

```
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
```

spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming

RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                     "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
               "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }

  },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

      {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
```

```
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
         "type": "structure",
         "members":
{
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                  "smithy.api#required": {}
               }
            },
            "LicenseServerSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                    "smithy.api#required": {}
```

```
    }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

    },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
```

```
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

                 "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }

            },
            "FailureMessage": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
        },
        "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
      "type": "structure",
      "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        },
        "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                "smithy.api#required": {}
            }
        },
```

```
            "LastStatusCheckDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
          },
          "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
```

```
            },
            "DisassociationDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
         "type": "list",
         "member": {

     "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
            },
            {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
      },
      {

        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
      },
      "aws.protocols#restJson1": {},
      "smithy.api#cors": {
        "additionalAllowedHeaders": [
          "*",
          "content-type",
          "x-amz-content-sha256",
          "x-amz-user-agent",
          "x-amzn-platform-id",
          "x-amzn-trace-id",
          "content-length",
           "x-api-key",
          "authorization",
          "x-amz-date",
          "x-amz-security-token",
          "Access-Control-Allow-Headers",
          "Access-Control-Allow-Methods",
          "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
          "x-amzn-errortype",
```

```
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
```

```
                "ref": "Endpoint"
              }
            ]

  }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
          },
          {
            "conditions": [

{
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
          },
          {
            "conditions": [],
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },

      "properties": {},
                "headers": {}
              },
```

```
                "type": "endpoint"
            }
          ],
          "type": "tree"
        },
        {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Region"
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
```

```
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

"fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    },

                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
```

```
                "headers": {}
              },
              "type": "endpoint"
            }

        ],
          "type": "tree"
        },
        {
          "conditions": [],
          "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
          "type": "error"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [

    {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "fn": "getAttr",
                  "argv": [

        {
                      "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                  ]
                },
                true
              ]
```

```
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {

    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
```

```
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]

            }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    }
],
```

```
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ]
          },
          "smithy.rules#endpointTests": {
            "testCases": [
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",

            "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                  }
                },
                "params": {
                  "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

    "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
 true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
```

```
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
```

```
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
              }
            },
            "params": {
              "Region": "us-iso-east-1",
              "UseFIPS":
false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {

      "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
              "endpoint": {
                "url": "https://example.com"
              }
            },
            "params": {
              "Region": "us-east-1",
```

```
                "UseFIPS": false,

        "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
               }
            },
            {
              "documentation": "Missing region",
              "expect": {
```

```
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
     }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
```

```
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
```

```
        },
        "LicenseServers": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
          }
        },
        "CreationTime": {
          "target": "smithy.api#Timestamp",

    "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
```

```
                "value": "PROVISIONED",
                "documentation": ""
              },
              {
                "name": "DELETING",
                "value": "DELETING",
                "documentation": ""
              },
              {
                "name": "DELETION_FAILED",
                "value": "DELETION_FAILED",
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
            }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

      {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
```

```
operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

},
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
```

```
            "ResourceArn": {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                  "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
         "type": "structure",
         "members": {
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}

 }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
```

```
    "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output":
 {}
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS) license server.</p>"
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
```

```
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

          },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                  "smithy.api#required": {}
                }
              }
          },
          "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
              "smithy.api#length": {
                "min": 1
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
 {
        "type": "string",
        "traits": {
          "smithy.api#length": {
            "min": 5,
            "max": 200
```

```
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
```

```
{
        "type": "structure",
        "members": {
           "message": {
              "target": "smithy.api#String"
           }
        },
        "traits": {
           "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
           "smithy.api#error": "client"
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
           "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                 "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                 "smithy.api#length": {
                    "min": 1
                 },
                 "smithy.api#required": {}
              }

     },
           "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                 "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                 "smithy.api#required": {}
              }
           }
        },
        "traits": {
           "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
     },
     "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
           "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        },
        "IdentityProvider":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
           }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }

        }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {

    "min": 0,
            "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

            }
            },
            "TagKeys": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
        "smithy.api#http": {

  "code": 200,
          "method": "POST",
          "uri": "/identity-provider/UpdateIdentityProviderSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
```

```
            },
            "IdentityProviderArn":
 {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }

            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
```

```
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
```

```
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
```

```
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string
```

```
 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
```

```
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}
```

```
type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
```

```
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
```

```
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string
```

```
 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
```

```
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
```

```
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
```

```
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
```

```
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
           "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of name to filter by.</p>"
           }
        },
        "Values": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
          }
        },
        "Operator": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
          "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
           },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
              "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
```

```
        "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
```

```
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
          },
          "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
              "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
          },
          "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
          },
          "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
          },
```

```
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
          }
        ],
        "traits": {
          "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
          },
          "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
```

```
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },
                        true
                      ]
                    }

        ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]

}
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
```

```
                {
                  "conditions": [
                    {

              "fn": "aws.partition",
                      "argv": [
                        {
                          "ref": "Region"
                        }
                      ],
                      "assign": "PartitionResult"
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {

                        "ref": "UseFIPS"
                            },
                            true
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]
                        }

],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {
```

```
                                        "ref": "PartitionResult"

                    },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [

                                     {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {

      "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

            "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled
but this partition does not support FIPS",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
```

```
                                        "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [

        {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"

                }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
  }
 ],
 "type": "tree"
 }
 ]
},
"smithy.rules#endpointTests": {
 "testCases": [
  {
   "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
   "expect": {
    "endpoint": {
     "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

 }
   },
   "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
   }
  },
  {
   "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
   "expect": {
    "endpoint": {
     "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
   },
   "params": {
    "Region": "us-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

    },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
```

```
"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                      }
                     },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": true
                    }
                  },
                  {
                    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                      }
                    },
                    "params": {
                      "Region": "cn-north-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
              }
            },
            "params": {
              "Region": "us-isob-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
              "Region": "us-isob-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
```

```
                    "expect": {
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": false,
                        "UseDualStack": true,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "Missing region",

                "expect": {
                        "error": "Invalid Configuration: Missing Region"
                    }
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
```

```
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
```

```
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
```

```
"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                    "min": 1,
                    "max": 16384
```

```
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
 isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
 next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
          },
          "output": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
 return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
 nextToken from a previously truncated response.</p>"
              }
            }
          },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "EQUAL",
                "value": "Equal",
```

```
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
          "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
                }
            },
            "SubscriptionProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
```

```
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
               },
              {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
              },
              {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
              },
              {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
              }
            ]
```

```
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
                }
```

```
            },
            "InstanceCount": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
               "traits": {
                  "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
         "type": "list",

       "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "RedHat",
                  "value": "RedHat",
                  "documentation": "RedHat subscription provider namespace"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
         "type":
 "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
         "type": "string",
         "traits": {
```

```
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
```

```
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
 {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
```

```
Region.</p>"
                }
            }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }

    },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
```

```
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.55 bash 5.2.37-r0

1.55.1 Available under license :

Shellfloat is copyright (c) 2020 by Michael Wood.

==========

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work
released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code.
And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed
this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying,

distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source

includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met.
This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your

work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no

further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the
only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied

by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or

author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual
assumptions of liability to the recipient, for
any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional
terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation
by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept
this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if

the predecessor
has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling
its contributor version,
but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent
license,
and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner

consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM

IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM
TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file
to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper
mail.

If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<https://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine
library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License. But first, please read
<https://www.gnu.org/licenses/why-not-lgpl.html>.

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to

assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of "copyleft", which means that derivative
works of the document must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We
have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that
contains a notice placed by the copyright holder saying it can be
distributed under the terms of this License. Such a notice grants a
world-wide, royalty-free license, unlimited in duration, to use that
work under the conditions stated herein. The "Document", below,
refers to any such manual or work. Any member of the public is a
licensee, and is addressed as "you". You accept the license if you
copy,
modify or distribute the work in a way requiring permission
under copyright law.

A "Modified Version" of the Document means any work containing the
Document or a portion of it, either copied verbatim, or with
modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of
the Document that deals exclusively with the relationship of the
publishers or authors of the Document to the Document's overall
subject (or to related matters) and contains nothing that could fall
directly within that overall subject. (Thus, if the Document is in
part a textbook of mathematics, a Secondary Section may not explain
any mathematics.) The relationship could be a matter of historical
connection with the subject or with related matters, or of legal,
commercial, philosophical, ethical or political position regarding
them.

The "Invariant Sections" are certain Secondary Sections whose titles
are designated, as being those of Invariant Sections,
 in the notice
that says that the Document is released under this License. If a
section does not fit the above definition of Secondary then it is not
allowed to be designated as Invariant. The Document may contain zero
Invariant Sections. If the Document does not identify any Invariant
Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed,
as Front-Cover Texts or Back-Cover Texts, in the notice that says that
the Document is released under this License. A Front-Cover Text may
be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy,
represented in a format whose specification is available to the
general public, that is suitable for revising the document
straightforwardly with generic text editors or (for images composed of
pixels) generic paint programs or (for drawings) some widely available
drawing editor, and that is suitable for input to text formatters
 or
for automatic translation to a variety of formats suitable for input
to text formatters. A copy made in an otherwise Transparent file
format whose markup, or absence of markup, has been arranged to thwart
or discourage subsequent modification by readers is not Transparent.
An image format is not Transparent if used for any substantial amount
of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain
ASCII without markup, Texinfo input format, LaTeX input format, SGML
or XML using a publicly available DTD, and standard-conforming simple
HTML, PostScript or PDF designed for human modification. Examples of
transparent image formats include PNG, XCF and JPG. Opaque formats
include proprietary formats that can be read and edited only by
proprietary word processors, SGML or XML for which the DTD and/or
processing tools are not generally available, and the
machine-generated HTML, PostScript or PDF produced by some
 word
processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself,
plus such following pages as are needed to hold, legibly, the material
this License requires to appear in the title page. For works in
formats which do not have any title page as such, "Title Page" means
the text near the most prominent appearance of the work's title,
preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The
Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section
3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve

the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five),

unless they release you from this requirement.
C. State on the Title page the name of the publisher of the
Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications
adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice
giving the public permission to use the Modified Version under the
terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections
and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add
to it an item stating at least the title, year,
new authors, and
publisher of the Modified Version as given on the Title Page. If
there is no section Entitled "History" in the Document, create one
stating the title, year, authors, and publisher of the Document as
given on its Title Page, then add an item describing the Modified
Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for
public access to a Transparent copy of the Document, and likewise
the network locations given in the Document for previous versions
it was based on. These may be placed in the "History" section.
You may omit a network location for a work that was published at
least four years before the Document itself, or if the original
publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications",
Preserve the Title of the section, and preserve in the section all
the substance and tone of each of the contributor
acknowledgements
and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document,
unaltered in their text and in their titles. Section numbers
or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section
may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements"
or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document, you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but
endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History"
in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives
with other separate
and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under
the terms of section 4.
Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the

Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See

http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and
license notices just after the title page:

  Copyright (c) YEAR YOUR NAME.
  Permission is granted to copy, distribute and/or modify this document
   under the terms of the GNU Free Documentation License, Version 1.3
  or any later version published by the Free Software Foundation;
  with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
  A copy of the license is included in the section entitled "GNU
  Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the "with...Texts." line with this:

  with the Invariant Sections being LIST THEIR TITLES, with the
  Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed

to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which
contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the
source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each
time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit royalty-free redistribution
of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through
any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number
of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN
IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w'
and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
@c The GNU Free Documentation License.
@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,
@c hence no sectioning command or @node.

@display
Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
@end display

@enumerate 0
@item
PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document @dfn{free} in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative
works of the document
 must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

@item
APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that
contains a notice placed by the copyright holder saying it can be
distributed under the terms of this License. Such a notice grants a
world-wide, royalty-free license, unlimited in duration, to use that
work under the conditions stated herein. The ``Document'', below,
refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent'' copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent'' is called ``Opaque''.

Examples of suitable formats for Transparent copies include plain
@sc{ascii} without markup, Texinfo input format, La@TeX{} input
format, @acronym{SGML} or @acronym{XML} using a publicly available
@acronym{DTD}, and standard-conforming simple @acronym{HTML},
PostScript or @acronym{PDF} designed for human modification. Examples
of transparent image formats include @acronym{PNG}, @acronym{XCF} and
@acronym{JPG}. Opaque formats
 include proprietary formats that can be
read and edited only by proprietary word processors, @acronym{SGML} or
@acronym{XML} for which the @acronym{DTD} and/or processing tools are
not generally available, and the machine-generated @acronym{HTML},
PostScript or @acronym{PDF} produced by some word processors for
output purposes only.

The ``Title Page'' means, for a printed book, the title page itself,
plus such following pages as are needed to hold, legibly, the material
this License requires to appear in the title page. For works in
formats which do not have any title page as such, ``Title Page'' means
the text near the most prominent appearance of the work's title,
preceding the beginning of the body of the text.

The ``publisher'' means any person or entity that distributes copies
of the Document to the public.

A section ``Entitled XYZ'' means a named subunit of the Document whose
title either is precisely XYZ or contains XYZ in parentheses following
text that translates XYZ in
 another language. (Here XYZ stands for a
specific section name mentioned below, such as ``Acknowledgements'',
``Dedications'', ``Endorsements'', or ``History''.)  To ``Preserve the Title''
of such a section when you modify the Document means that it remains a
section ``Entitled XYZ'' according to this definition.

The Document may include Warranty Disclaimers next to the notice which
states that this License applies to the Document. These Warranty
Disclaimers are considered to be included by reference in this
License, but only as regards disclaiming warranties: any other
implication that these Warranty Disclaimers may have is void and has
no effect on the meaning of this License.

@item
VERBATIM COPYING

You may copy and distribute the Document in any medium, either
commercially or noncommercially, provided that this License, the
copyright notices, and the license notice saying this License applies
to the Document are reproduced in all copies, and that you add no other
conditions whatsoever

to those of this License. You may not use
technical measures to obstruct or control the reading or further
copying of the copies you make or distribute. However, you may accept
compensation in exchange for copies. If you distribute a large enough
number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and
you may publicly display copies.

@item
COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have
printed covers) of the Document, numbering more than 100, and the
Document's license notice requires Cover Texts, you must enclose the
copies in covers that carry, clearly and legibly, all these Cover
Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on
the back cover. Both covers must also clearly and legibly identify
you as the publisher of these copies. The front cover must present
the full title with all words of the title equally prominent and
visible.
You may add other material on the covers in addition.
Copying with changes limited to the covers, as long as they preserve
the title of the Document and satisfy these conditions, can be treated
as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit
legibly, you should put the first ones listed (as many as fit
reasonably) on the actual cover, and continue the rest onto adjacent
pages.

If you publish or distribute Opaque copies of the Document numbering
more than 100, you must either include a machine-readable Transparent
copy along with each Opaque copy, or state in or with each Opaque copy
a computer-network location from which the general network-using
public has access to download using public-standard network protocols
a complete Transparent copy of the Document, free of added material.
If you use the latter option, you must take reasonably prudent steps,
when you begin distribution of Opaque copies in quantity, to ensure
that
this Transparent copy will remain thus accessible at the stated
location until at least one year after the last time you distribute an
Opaque copy (directly or through your agents or retailers) of that
edition to the public.

It is requested, but not required, that you contact the authors of the
Document well before redistributing any large number of copies, to give
them a chance to provide you with an updated version of the Document.

@item
MODIFICATIONS

You may copy and distribute a Modified Version of the Document under
the conditions of sections 2 and 3 above, provided that you release
the Modified Version under precisely this License, with the Modified
Version filling the role of the Document, thus licensing distribution
and modification of the Modified Version to whoever possesses a copy
of it. In addition, you must do these things in the Modified Version:

@enumerate A
@item
Use in the Title Page (and on the covers, if any) a title distinct
from that of the Document, and from
 those of previous versions
(which should, if there were any, be listed in the History section
of the Document). You may use the same title as a previous version
if the original publisher of that version gives permission.

@item
List on the Title Page, as authors, one or more persons or entities
responsible for authorship of the modifications in the Modified
Version, together with at least five of the principal authors of the
Document (all of its principal authors, if it has fewer than five),
unless they release you from this requirement.

@item
State on the Title page the name of the publisher of the
Modified Version, as the publisher.

@item
Preserve all the copyright notices of the Document.

@item
Add an appropriate copyright notice for your modifications
adjacent to the other copyright notices.

@item
Include, immediately after the copyright notices, a license notice
giving the public permission to use the Modified Version under the
terms of this License, in the form shown in
 the Addendum below.

@item
Preserve in that license notice the full lists of Invariant Sections
and required Cover Texts given in the Document's license notice.

@item
Include an unaltered copy of this License.

@item
Preserve the section Entitled ``History'', Preserve its Title, and add
to it an item stating at least the title, year, new authors, and
publisher of the Modified Version as given on the Title Page. If
there is no section Entitled ``History'' in the Document, create one
stating the title, year, authors, and publisher of the Document as
given on its Title Page, then add an item describing the Modified
Version as stated in the previous sentence.

@item
Preserve the network location, if any, given in the Document for
public access to a Transparent copy of the Document, and likewise
the network locations given in the Document for previous versions
it was based on. These may be placed in the ``History'' section.
You may omit a network location for a work that was published at
least
four years before the Document itself, or if the original
publisher of the version it refers to gives permission.

@item
For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve
the Title of the section, and preserve in the section all the
substance and tone of each of the contributor acknowledgements and/or
dedications given therein.

@item
Preserve all the Invariant Sections of the Document,
unaltered in their text and in their titles. Section numbers
or the equivalent are not considered part of the section titles.

@item
Delete any section Entitled ``Endorsements''. Such a section
may not be included in the Modified Version.

@item
Do not retitle any existing section to be Entitled ``Endorsements'' or
to conflict in title with any Invariant Section.

@item
Preserve any Warranty Disclaimers.
@end enumerate

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document,

you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements", provided it contains
nothing but endorsements of your Modified Version by various
parties---for example, statements of peer review or that the text has
been approved by an organization as the authoritative definition of a
standard.

You may add a passage of up to five words as a Front-Cover Text, and a
passage of up to 25 words as a Back-Cover Text, to the end of the list
of Cover Texts in the Modified Version. Only one passage of
Front-Cover Text and one of Back-Cover Text may be added by (or
through arrangements made by) any one entity. If the Document already
includes a cover text for the same cover, previously added by you or
by arrangement made by the same entity you are acting on behalf
of,
you may not add another; but you may replace the old one, on explicit
permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License
give permission to use their names for publicity for or to assert or
imply endorsement of any Modified Version.

@item
COMBINING DOCUMENTS

You may combine the Document with other documents released under this
License, under the terms defined in section 4 above for modified
versions, provided that you include in the combination all of the
Invariant Sections of all of the original documents, unmodified, and
list them all as Invariant Sections of your combined work in its
license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and
multiple identical Invariant Sections may be replaced with a single
copy. If there are multiple Invariant Sections with the same name but
different contents, make the title of
each such section unique by
adding at the end of it, in parentheses, the name of the original
author or publisher of that section if known, or else a unique number.
Make the same adjustment to the section titles in the list of
Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History"
in the various original documents, forming one section Entitled

``History''; likewise combine any sections Entitled ``Acknowledgements'',
and any sections Entitled ``Dedications''. You must delete all
sections Entitled ``Endorsements.''

@item
COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents
released under this License, and replace the individual copies of this
License in the various documents with a single copy that is included in
the collection, provided that you follow the rules of this License for
verbatim copying of each of the documents in all other respects.

You may extract
 a single document from such a collection, and distribute
it individually under this License, provided you insert a copy of this
License into the extracted document, and follow this License in all
other respects regarding verbatim copying of that document.

@item
AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate
and independent documents or works, in or on a volume of a storage or
distribution medium, is called an ``aggregate'' if the copyright
resulting from the compilation is not used to limit the legal rights
of the compilation's users beyond what the individual works permit.
When the Document is included in an aggregate, this License does not
apply to the other works in the aggregate which are not themselves
derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these
copies of the Document, then if the Document is less than one half of
the entire aggregate, the Document's Cover Texts may
 be placed on
covers that bracket the Document within the aggregate, or the
electronic equivalent of covers if the Document is in electronic form.
Otherwise they must appear on printed covers that bracket the whole
aggregate.

@item
TRANSLATION

Translation is considered a kind of modification, so you may
distribute translations of the Document under the terms of section 4.
Replacing Invariant Sections with translations requires special
permission from their copyright holders, but you may include
translations of some or all Invariant Sections in addition to the

original versions of these Invariant Sections. You may include a
translation of this License, and all the license notices in the
Document, and any Warranty Disclaimers, provided that you also include
the original English version of this License and the original versions
of those notices and disclaimers. In case of a disagreement between
the translation and the original version of this License or a notice
or disclaimer, the original
 version will prevail.

If a section in the Document is Entitled ``Acknowledgements'',
``Dedications'', or ``History'', the requirement (section 4) to Preserve
its Title (section 1) will typically require changing the actual
title.

@item
TERMINATION

You may not copy, modify, sublicense, or distribute the Document
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense, or distribute it is void, and
will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license
from a particular copyright holder is reinstated (a) provisionally,
unless and until the copyright holder explicitly and finally
terminates your license, and (b) permanently, if the copyright holder
fails to notify you of the violation by some reasonable means prior to
60 days after the cessation.

Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies
 you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License. If your rights have been terminated and not permanently
reinstated, receipt of a copy of some or all of the same material does
not give you any rights to use it.

@item
FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions
of the GNU Free Documentation License from time to time. Such new

versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns. See
@uref{http://www.gnu.org/copyleft/}.

Each version of the License is given a distinguishing version number.
If the Document specifies that
 a particular numbered version of this
License ``or any later version'' applies to it, you have the option of
following the terms and conditions either of that specified version or
of any later version that has been published (not as a draft) by the
Free Software Foundation. If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation. If the Document
specifies that a proxy can decide which future versions of this
License can be used, that proxy's public statement of acceptance of a
version permanently authorizes you to choose that version for the
Document.

@item
RELICENSING

``Massive Multiauthor Collaboration Site'' (or ``MMC Site'') means any
World Wide Web server that publishes copyrightable works and also
provides prominent facilities for anybody to edit those works. A
public wiki that anybody can edit is an example of such a server. A
``Massive Multiauthor Collaboration'' (or ``MMC'')
 contained in the
site means any set of copyrightable works thus published on the MMC
site.

``CC-BY-SA'' means the Creative Commons Attribution-Share Alike 3.0
license published by Creative Commons Corporation, a not-for-profit
corporation with a principal place of business in San Francisco,
California, as well as future copyleft versions of that license
published by that same organization.

``Incorporate'' means to publish or republish a Document, in whole or
in part, as part of another Document.

An MMC is ``eligible for relicensing'' if it is licensed under this
License, and if all works that were first published under this License
somewhere other than this MMC, and subsequently incorporated in whole
or in part into the MMC, (1) had no cover texts or invariant sections,
and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site
under CC-BY-SA on the same site at any time before August 1, 2009,
provided the

MMC is eligible for relicensing.

@end enumerate

@page
@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and
license notices just after the title page:

@smallexample
@group
 Copyright (C) @var{year} @var{your name}.
 Permission is granted to copy, distribute and/or modify this document
 under the terms of the GNU Free Documentation License, Version 1.3
 or any later version published by the Free Software Foundation;
 with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
 Texts. A copy of the license is included in the section entitled ``GNU
 Free Documentation License".
@end group
@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the ``with@dots{}Texts." line with this:

@smallexample
@group
   with the Invariant Sections being @var{list their titles}, with
   the Front-Cover
 Texts being @var{list}, and with the Back-Cover Texts
   being @var{list}.
@end group
@end smallexample

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.

@c Local Variables:
@c ispell-local-pdict: "ispell-dict"
@c End:

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to
freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1)
assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as

changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand
ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees"
and
"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without

permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally

available free
programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered
by this License only if the output, given its
content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;
section 10
makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such
measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately
publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.

b) The work must carry prominent notices stating that it is
released under this License and any conditions added under section
7. This requirement modifies the requirement in section
4 to
"keep intact all notices".

c) You must license the entire work, as a whole, under this
License to anyone who comes into possession of a copy. This
License will therefore apply, along with any applicable section 7
additional terms, to the whole of the work, and all its parts,
regardless of how they are packaged. This License gives no
permission to license the work in any other way, but it does not
invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated

place (gratis or for a charge), and offer equivalent access to the Corresponding Source in
the same way through the same place at no
further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source
as a System Library, need not be
included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization
keys, or other information required to install
and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the

User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in
ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the
entire Program shall
be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or

limiting liability differently from the
terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or
author attributions in that material or in the Appropriate Legal
Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or
requiring that modified versions of such material be marked in
reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or
authors of the material; or

e) Declining to grant rights under trademark law for use of some
trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that
material by anyone who conveys the material (or modified versions of
it) with contractual assumptions of liability to the recipient, for
any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10. If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term. If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above
requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly
provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright
holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights
granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For
purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients. "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered
work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License. You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or
that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.
Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to
do, use the GNU Lesser General
Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
Unless otherwise stated, all files in this directory are Copyright (C)
1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,
2004,2005,2006,2007,2008,2009,2010,2011
Free Software Foundation, Inc.

See the file COPYING in the bash distribution root directory for copying
and usage restrictions.

The file ifs-posix.tests is Copyright (C) 2005 Glen Fowler.

1.56 go.yaml.in-yaml 2.4.3

1.56.1 Available under license :

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.57 aws-sdk-go-v2-credentials 1.18.19

1.57.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```
 spanInitializeEnd
Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}
```

```
type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
```

ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

```
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
```

```
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
  }
 }
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

```
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
```

```
//
//   - Fingerprint
//
//   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
```

```
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
        "type": "structure",
        "members": {
          "DirectoryId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
            "traits": {
              "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
          },
          "ActiveDirectorySettings":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

           "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
```

```
            },
            "DomainIpv4List": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                  "smithy.api#length": {
                     "min": 1,
                     "max": 2
                  }
               }
            },
            "DomainIpv6List": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                  "smithy.api#length": {
                     "min": 1,
                     "max": 2
                  }
               }
            },
            "DomainCredentialsProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
               "traits": {
                  "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
               }
            },
            "DomainNetworkSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
```

```
                "name": "SELF_MANAGED",
                 "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
              },
              {
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Arn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
          },
          "output": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
```

```
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
    }
```

```
},
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

      {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/CreateLicenseServerEndpoint"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
```

```
{
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                  "smithy.api#required": {}
               }
            },
            "LicenseServerSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
               }
            },
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
               }
            }
```

```
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
              "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

             },
              "smithy.api#idempotent": {}
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
            "type": "structure",
            "members": {
              "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                  "smithy.api#required": {}
                }
              },
              "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                  "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                  "smithy.api#required": {}

    }
                }
              },
              "traits": {
                "smithy.api#input": {}
              }
            },
            "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
              "type": "structure",
              "members": {
                "LicenseServerEndpoint": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
                  "traits": {
                    "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
```

```
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

    },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },

    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
              }
            },
            "Value": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
                "smithy.api#required": {}
              }
```

```
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
            "smithy.api#required": {}
          }
        },
        "Products": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
          }
        },
        "LastStatusCheckDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                "smithy.api#required": {}
            }
```

```
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {

                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
```

```
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
```

```
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
 },
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
 },
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
```

```
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
```

```
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
```

```
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
        },
        {
            "conditions": [],
            "endpoint": {
                "url": {
                    "ref": "Endpoint"
                },

        "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
```

```
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
               "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },

true
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {

"fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
```

```
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            },

            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }

            ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }

        ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
            "type": "error"
        }
```

```
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [

{
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                    {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
```

```
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"

}
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {

    "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]

     }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
```

```
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                    "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
```

```
"expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",

              "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

  "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
```

```
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
```

```
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,

         "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
```

```
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server.</p>"
            }
        },
        "HealthStatus": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>"
            }
        },
```

```
        "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
            }
        },
        "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
        },
        "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
        },
        "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
        },
        "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
```

```
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }

  }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
```

```
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
             "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
             }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
```

```
criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
             }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",
```

```
    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
                }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
         "type": "structure",
         "members": {
            "ProductUserSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
```

```
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
        },
        "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
        },
        "SubscriptionEndDate": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
                }
```

```
        },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
```

```
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
          "type": "structure",
          "members": {
            "SecretId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                  "min": 1
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 5,
              "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
          "type": "structure",
          "members": {
            "Endpoint": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
              }
            }
          },
```

```
        "traits": {
          "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

       "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
```

```
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
 {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
```

```
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }

         }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StringList": {
          "type": "list",
```

```
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

"min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
```

```
        ],
        "traits": {
          "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
```

```
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

            }
            },
            "TagKeys": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
              "method": "POST",
              "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "IdentityProviderArn":
 {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
              }
            },
            "UpdateSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
              "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related
```

configuration settings. You can update any combination of settings in a single operation such as the: <ul> Subnets which you want to add to provision VPC endpoints.   Subnets which you want to remove the VPC endpoints from.   Security group ID which permits traffic to the VPC endpoints. ",
                "smithy.api#required": {}
            }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": " The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers. ",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": " The ID of one or more subnets to remove. ",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",

```
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
```

```
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
```

```
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region:
      region,
 ServiceID:     ServiceID,
 OperationName: "GetLicense",
```

```
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
```

```
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
```

```
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string
```

```
// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata
```

```
// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
```

```
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options)) (*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
```

```
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
```

```
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
```

```
//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string
```

```
 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
         "type": "structure",
         "members": {
            "SubscriptionProviderArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
         "type": "structure",
         "members": {},
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
         "type": "structure",
         "members": {
            "Name": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The type of name to filter by.</p>"
               }
            },
```

```
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
              "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
          "type":
 "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
          },
          "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
```

```
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
           "traits": {
             "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
```

```
            },
            "UsageOperation": {
               "target": "smithy.api#String",

   "traits": {
                  "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
               }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
            },
            "LastUpdatedTime": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
               }
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
          },
          "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
```

```
"operations": [
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
         "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
```

```
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
```

```
        {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Endpoint"
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
```

```
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

        "fn": "aws.partition",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ],
                    "assign": "PartitionResult"
                  }
                ],
                "rules": [
                  {
                    "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {

                "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }

    ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"

            },
                                "supportsFIPS"
                            ]
                          }
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
```

```
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {

        "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
```

```
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

                "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true

                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
```

```
                "conditions": [],
                "error": "FIPS is enabled
but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true

        ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {

                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [

{
                        "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
```

```
                "type": "error"
            }
          ],
          "type": "tree"
        }
      ]
    },
    "smithy.rules#endpointTests": {
      "testCases": [
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

      }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

       "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
```

```
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
          }

    },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {

  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
```

```
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }

       },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
```

```
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
```

```
      }
    },

  {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

   "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
```

```
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
 <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
```

```
streamline results:</p>\n <ul>\n <li>\n <p>\n <code>AccountID</code>\n
</p>\n </li>\n <li>\n <p>\n <code>AmiID</code>\n </p>\n </li>\n
<li>\n <p>\n <code>DualSubscription</code>\n </p>\n </li>\n
<li>\n <p>\n <code>InstanceID</code>\n </p>\n </li>\n <li>\n
<p>\n <code>InstanceType</code>\n </p>\n </li>\n <li>\n <p>\n
<code>ProductCode</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Region</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Status</code>\n </p>\n </li>\n <li>\n <p>\n
<code>UsageOperation</code>\n </p>\n </li>\n </ul>\n <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n <ul>\n
<li>\n <p>\n <code>contains</code>\n </p>\n </li>\n <li>\n
<p>\n <code>equals</code>\n </p>\n </li>\n <li>\n
<p>\n <code>Notequal</code>\n </p>\n </li>\n </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 16384
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
```

```
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n       <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
```

```
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"

          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListRegisteredSubscriptionProviders"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "RegisteredSubscriptionProviders"
        }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
```

```
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "tags": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "EQUAL",
            "value": "Equal",
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {

"name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },

     {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
```

```
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
               "traits": {
                 "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                 "smithy.api#required": {}
               }
            },
            "SecretArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
              "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've
stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat
Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                 "smithy.api#required": {}
               }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
               }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                 "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
```

```
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
            }
          },
           "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
```

```
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
```

```
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
    {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
    resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
```

```
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
```

```
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
```

```
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
{}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
```

```
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
                }
             }
          }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
              "message": {
                  "target": "smithy.api#String"
              }
          },
          "traits": {
              "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
              "smithy.api#error": "client"
          }
      }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.58 pam 1.7.0-r4

1.58.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed

to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work
which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably
considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code
or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy
the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.
Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit royalty-free
redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through
any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version
number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE
LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the
License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands
`show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

```
# Automatically generated by apkbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
  INSTALLDIRS=vendor \
  NO_PACKLIST=1 \
  NO_PERLLOCAL=1
 make
}

check() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 make test
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
  BSD-Resource-1.2911.tar.gz
"
# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
```

```
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
 0001-bsd-games-scores-dir.patch
 "
options="!check" # no tests

build() {
 # some GNU autoconf options are ignored, but it works
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr \
  --sysconfdir=/etc \
  --mandir=/usr/share/man \
  --scoredir=/usr/share/$pkgname
 make
}

package() {
 make DESTDIR="$pkgdir" install
 install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
  bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6  0001-bsd-games-scores-dir.patch
"
```

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
 WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2
 "

builddir="$srcdir"/libart_lgpl-$pkgver

prepare() {
 update_config_sub
 default_prepare
}

build() {
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr
 make
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
Contributor: Patrycja Rosa <alpine@ptrcnull.me>
Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
 cdefs.h
 queue.h
 tree.h
"
builddir="$srcdir"
options="!check" # just headers

package() {
 mkdir -p "$pkgdir"
 install -Dm644 -t "$pkgdir"/usr/include/sys \
  cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h

```
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3  tree.h
"
# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>
# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>
pkgname=spdx-licenses
pkgver=3.26.0
pkgrel=0
pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdfnt rdfturtle rdfxml template text"

for type in $_types; do
 subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
 mkdir -p "$pkgdir"
}

_subpkg() {
 local type=${subpkgname/$pkgname-/}
 pkgdesc="$pkgdesc ($type)"
 install_if="$pkgname=$pkgver-r$pkgrel"
 mkdir -p "$subpkgdir"/usr/share/spdx
 cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}

list() {
 pkgdesc="$pkgdesc (licence list)"
 mkdir -p "$subpkgdir"/usr/share/spdx
 local i; for i in "$builddir"/text/*.txt;
 do
  local license=${i##*/}
  echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
 done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a  license-list-data-3.26.0.tar.gz
```

"
Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)
The text rotation code in w_rottext.c was written by Alan Richardson.
The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker
The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick
The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:
"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95
http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html
Contributor: Dhruvin Gandhi <contact@dhru.vin>
Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=py3-license-expression
_pkgname=license_expression
pkgver=30.4.1
pkgrel=0
pkgdesc="Library to parse, compare, simplify and normalize license expressions"
url="https://github.com/nexB/license-expression"
arch="noarch"
license="Apache-2.0"
depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir/$_pkgname-$pkgver"

build() {

```
 gpep517 build-wheel \
  --wheel-dir .dist \
  --output-fd 3 3>&1 >&2
}

check() {
 python3 -m venv --clear --without-pip --system-site-packages .testenv
 .testenv/bin/python3 -m installer .dist/*.whl
 .testenv/bin/python3 -m pytest
}

package() {
 python3 -m installer -d "$pkgdir" \
  .dist/*.whl
}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
  license_expression-30.4.1.tar.gz
"
/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
* Copyright (c) 1991, 1993
* The Regents of the University of California.  All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
```

```
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
*/

#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_

/*
* This file defines five types of data structures: singly-linked lists,
* lists, simple queues, tail queues, and circular queues.
*
* A singly-linked list is headed by a single forward pointer. The
* elements are singly linked for minimum space and
 pointer manipulation
* overhead at the expense of O(n) removal for arbitrary elements. New
* elements can be added to the list after an existing element or at the
* head of the list.  Elements being removed from the head of the list
* should use the explicit macro for this purpose for optimum
* efficiency. A singly-linked list may only be traversed in the forward
* direction.  Singly-linked lists are ideal for applications with large
* datasets and few or no removals or for implementing a LIFO queue.
*
* A list is headed by a single forward pointer (or an array of forward
* pointers for a hash table header). The elements are doubly linked
* so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before
* or after an existing element or at the head of the list. A list
* may only be traversed in the forward direction.
*
* A simple queue is headed by a pair of pointers, one the head of the
* list and the
 other to the tail of the list. The elements are singly
* linked to save space, so elements can only be removed from the
* head of the list. New elements can be added to the list after
* an existing element, at the head of the list, or at the end of the
* list. A simple queue may only be traversed in the forward direction.
*
* A tail queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before or
* after an existing element, at the head of the list, or at the end of
* the list. A tail queue may be traversed in either direction.
*
* A circle queue is headed by a pair of pointers, one to the head of the
```

```
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements
 can be added to the list before or after
* an existing element, at the head of the list, or at the end of the list.
* A circle queue may be traversed in either direction, but has a more
* complex end of list detection.
*
* For details on the use of these macros, see the queue(3) manual page.
*/

/*
* Include the definition of NULL only on NetBSD because sys/null.h
* is not available elsewhere.  This conditional makes the header
* portable and it can simply be dropped verbatim into any system.
* The caveat is that on other systems some other header
* must provide NULL before the macros can be used.
*/
#ifdef __NetBSD__
#include <sys/null.h>
#endif

#if defined(QUEUEDEBUG)
# if defined(_KERNEL)
#  define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
#  include <err.h>
#  define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
* Singly-linked List definitions.
*/
#define SLIST_HEAD(name, type)      \
struct name {       \
 struct type *slh_first; /* first
 element */   \
}

#define SLIST_HEAD_INITIALIZER(head)     \
 { NULL }

#define SLIST_ENTRY(type)      \
struct {        \
 struct type *sle_next; /* next element */   \
}
```

```
/*
* Singly-linked List access methods.
*/
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head)  NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field)     \
 for((var) = (head)->slh_first;     \
    (var) != SLIST_END(head);     \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = SLIST_FIRST((head));    \
    (var) != SLIST_END(head) &&     \
    ((tvar) = SLIST_NEXT((var), field), 1);   \
    (var) = (tvar))

/*
* Singly-linked List functions.
*/
#define SLIST_INIT(head) do {      \
 (head)->slh_first = SLIST_END(head);    \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do {   \
 (elm)->field.sle_next = (slistelm)->field.sle_next;  \
 (slistelm)->field.sle_next
 = (elm);    \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do {   \
 (elm)->field.sle_next = (head)->slh_first;   \
 (head)->slh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_AFTER(slistelm, field) do {   \
 (slistelm)->field.sle_next =     \
    SLIST_NEXT(SLIST_NEXT((slistelm), field), field);  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do {    \
 (head)->slh_first = (head)->slh_first->field.sle_next;  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do {   \
 if ((head)->slh_first == (elm)) {    \
```

```
 SLIST_REMOVE_HEAD((head), field);   \
 }       \
 else {       \
  struct type *curelm = (head)->slh_first;  \
  while(curelm->field.sle_next != (elm))   \
   curelm = curelm->field.sle_next;  \
  curelm->field.sle_next =    \
      curelm->field.sle_next->field.sle_next;  \
 }       \
} while (/*CONSTCOND*/0)


/*
* List definitions.
 */
#define LIST_HEAD(name, type)      \
struct name {       \
 struct type *lh_first; /* first element */   \
}

#define LIST_HEAD_INITIALIZER(head)     \
 { NULL }

#define LIST_ENTRY(type)      \
struct {       \
 struct type *le_next; /* next element */   \
 struct type **le_prev; /* address of previous next element */ \
}

/*
* List access methods.
*/
#define LIST_FIRST(head)  ((head)->lh_first)
#define LIST_END(head)   NULL
#define LIST_EMPTY(head)  ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field)  ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field)     \
 for ((var) = ((head)->lh_first);    \
     (var) != LIST_END(head);     \
     (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = LIST_FIRST((head));    \
     (var) != LIST_END(head) &&     \
     ((tvar) = LIST_NEXT((var), field), 1);   \
     (var) = (tvar))
```

```
#define LIST_MOVE(head1, head2) do {     \
 LIST_INIT((head2));      \
 if (!LIST_EMPTY((head1)))
 {     \
  (head2)->lh_first = (head1)->lh_first;   \
  LIST_INIT((head1));     \
 }       \
} while (/*CONSTCOND*/0)

/*
* List functions.
*/
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)   \
 if ((head)->lh_first &&      \
    (head)->lh_first->field.le_prev != &(head)->lh_first) \
  QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
     __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field)     \
 if ((elm)->field.le_next &&     \
    (elm)->field.le_next->field.le_prev !=   \
    &(elm)->field.le_next)     \
  QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm),  \
     __FILE__, __LINE__);    \
 if (*(elm)->field.le_prev != (elm))    \
  QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm),  \
     __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)    \
 (elm)->field.le_next = (void *)1L;    \
 (elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,
 field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do {      \
 (head)->lh_first = LIST_END(head);    \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do {   \
 QUEUEDEBUG_LIST_OP((listelm), field)    \
 if (((elm)->field.le_next = (listelm)->field.le_next) !=  \
    LIST_END(head))      \
  (listelm)->field.le_next->field.le_prev =  \
     &(elm)->field.le_next;    \
 (listelm)->field.le_next = (elm);    \
```

```
 (elm)->field.le_prev = &(listelm)->field.le_next;  \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do {   \
 QUEUEDEBUG_LIST_OP((listelm), field)    \
 (elm)->field.le_prev = (listelm)->field.le_prev;  \
 (elm)->field.le_next = (listelm);    \
 *(listelm)->field.le_prev = (elm);    \
 (listelm)->field.le_prev = &(elm)->field.le_next;  \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do {    \
 QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field)  \
 if (((elm)->field.le_next
 = (head)->lh_first) != LIST_END(head))\
  (head)->lh_first->field.le_prev = &(elm)->field.le_next;\
 (head)->lh_first = (elm);     \
 (elm)->field.le_prev = &(head)->lh_first;   \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do {     \
 QUEUEDEBUG_LIST_OP((elm), field)    \
 if ((elm)->field.le_next != NULL)    \
  (elm)->field.le_next->field.le_prev =    \
      (elm)->field.le_prev;    \
 *(elm)->field.le_prev = (elm)->field.le_next;   \
 QUEUEDEBUG_LIST_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do {    \
 if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
  (elm2)->field.le_next->field.le_prev =   \
      &(elm2)->field.le_next;    \
 (elm2)->field.le_prev = (elm)->field.le_prev;   \
 *(elm2)->field.le_prev = (elm2);    \
 QUEUEDEBUG_LIST_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)

/*
* Simple queue definitions.
*/
#define SIMPLEQ_HEAD(name, type)     \
struct name {        \
 struct
 type *sqh_first; /* first element */   \
 struct type **sqh_last; /* addr of last next element */  \
}
```

```
#define SIMPLEQ_HEAD_INITIALIZER(head)    \
 { NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type)     \
struct {        \
 struct type *sqe_next; /* next element */   \
}

/*
* Simple queue access methods.
*/
#define SIMPLEQ_FIRST(head)  ((head)->sqh_first)
#define SIMPLEQ_END(head)  NULL
#define SIMPLEQ_EMPTY(head)  ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->sqh_first);    \
     (var) != SIMPLEQ_END(head);     \
     (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next)   \
 for ((var) = ((head)->sqh_first);    \
     (var) != SIMPLEQ_END(head) &&    \
     ((next = ((var)->field.sqe_next)), 1);   \
     (var) = (next))

/*
* Simple queue functions.
*/
#define SIMPLEQ_INIT(head) do {      \
 (head)->sqh_first = NULL;     \
 (head)->sqh_last
 = &(head)->sqh_first;    \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do {   \
 if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
  (head)->sqh_last = &(elm)->field.sqe_next;  \
 (head)->sqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do {   \
 (elm)->field.sqe_next = NULL;     \
 *(head)->sqh_last = (elm);     \
 (head)->sqh_last = &(elm)->field.sqe_next;   \
} while (/*CONSTCOND*/0)
```

```
#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL)\
  (head)->sqh_last = &(elm)->field.sqe_next;  \
 (listelm)->field.sqe_next = (elm);    \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do {    \
 if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
  (head)->sqh_last = &(head)->sqh_first;   \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do {   \
 if
 (((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
    == NULL)       \
  (head)->sqh_last = &(elm)->field.sqe_next;  \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do {   \
 if ((head)->sqh_first == (elm)) {    \
  SIMPLEQ_REMOVE_HEAD((head), field);   \
 } else {       \
  struct type *curelm = (head)->sqh_first;  \
  while (curelm->field.sqe_next != (elm))   \
   curelm = curelm->field.sqe_next;  \
  if ((curelm->field.sqe_next =    \
   curelm->field.sqe_next->field.sqe_next) == NULL) \
      (head)->sqh_last = &(curelm)->field.sqe_next; \
 }       \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do {    \
 if (!SIMPLEQ_EMPTY((head2))) {     \
  *(head1)->sqh_last = (head2)->sqh_first;  \
  (head1)->sqh_last = (head2)->sqh_last;  \
  SIMPLEQ_INIT((head2));     \
 }       \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field)     \
 (SIMPLEQ_EMPTY((head)) ?      \
  NULL :       \
        ((struct type
 *)(void *)    \
  ((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
* Tail queue definitions.
```

```
*/
#define _TAILQ_HEAD(name, type, qual)     \
struct name {        \
 qual type *tqh_first;  /* first element */  \
 qual type *qual *tqh_last; /* addr of last next element */ \
}
#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head)     \
 { TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual)     \
struct {        \
 qual type *tqe_next;  /* next element */  \
 qual type *qual *tqe_prev; /* address of previous next element */\
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
* Tail queue access methods.
*/
#define TAILQ_FIRST(head)  ((head)->tqh_first)
#define TAILQ_END(head)   (NULL)
#define TAILQ_NEXT(elm, field)  ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
 (*(((struct headname *)(void *)((head)->tqh_last))->tqh_last))
#define TAILQ_PREV(elm, headname, field) \
 (*(((struct headname
 *)(void *)((elm)->field.tqe_prev))->tqh_last))
#define TAILQ_EMPTY(head)  (TAILQ_FIRST(head) == TAILQ_END(head))


#define TAILQ_FOREACH(var, head, field)     \
 for ((var) = ((head)->tqh_first);    \
    (var) != TAILQ_END(head);     \
    (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next)   \
 for ((var) = ((head)->tqh_first);    \
    (var) != TAILQ_END(head) &&     \
    ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field)  \
 for ((var) = TAILQ_LAST((head), headname);   \
    (var) != TAILQ_END(head);     \
    (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
```

```
 for ((var) = TAILQ_LAST((head), headname);   \
    (var) != TAILQ_END(head) &&     \
    ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
* Tail queue functions.
*/
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head,
 elm, field)   \
 if ((head)->tqh_first &&     \
    (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
  QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)   \
 if (*(head)->tqh_last != NULL)     \
  QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field)     \
 if ((elm)->field.tqe_next &&     \
    (elm)->field.tqe_next->field.tqe_prev !=   \
    &(elm)->field.tqe_next)     \
  QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
      __FILE__, __LINE__);    \
 if (*(elm)->field.tqe_prev != (elm))    \
  QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
      __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)   \
 if ((elm)->field.tqe_next == NULL &&    \
    (head)->tqh_last != &(elm)->field.tqe_next)   \
  QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d",\
      (head), (elm), __FILE__,
 __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)    \
 (elm)->field.tqe_next = (void *)1L;    \
 (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do {      \
 (head)->tqh_first = TAILQ_END(head);    \
 (head)->tqh_last = &(head)->tqh_first;    \
} while (/*CONSTCOND*/0)
```

```
#define TAILQ_INSERT_HEAD(head, elm, field) do {	\
 QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field)	\
 if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head))\
  (head)->tqh_first->field.tqe_prev =	\
      &(elm)->field.tqe_next;	\
 else		\
  (head)->tqh_last = &(elm)->field.tqe_next;	\
 (head)->tqh_first = (elm);	\
 (elm)->field.tqe_prev = &(head)->tqh_first;	\
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head,
 elm, field) do {	\
 QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field)	\
 (elm)->field.tqe_next = TAILQ_END(head);	\
 (elm)->field.tqe_prev = (head)->tqh_last;	\
 *(head)->tqh_last = (elm);	\
 (head)->tqh_last = &(elm)->field.tqe_next;	\
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do {	\
 QUEUEDEBUG_TAILQ_OP((listelm), field)	\
 if (((elm)->field.tqe_next = (listelm)->field.tqe_next) !=	\
     TAILQ_END(head))		\
  (elm)->field.tqe_next->field.tqe_prev =	\
      &(elm)->field.tqe_next;	\
 else		\
  (head)->tqh_last = &(elm)->field.tqe_next;	\
 (listelm)->field.tqe_next = (elm);	\
 (elm)->field.tqe_prev = &(listelm)->field.tqe_next;	\
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do {	\
 QUEUEDEBUG_TAILQ_OP((listelm), field)	\
 (elm)->field.tqe_prev = (listelm)->field.tqe_prev;	\
 (elm)->field.tqe_next = (listelm);	\
 *(listelm)->field.tqe_prev = (elm);	\
 (listelm)->field.tqe_prev
 = &(elm)->field.tqe_next;	\
} while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do {	\
 QUEUEDEBUG_TAILQ_PREREMOVE((head), (elm), field)	\
 QUEUEDEBUG_TAILQ_OP((elm), field)	\
 if (((elm)->field.tqe_next) != TAILQ_END(head))	\
  (elm)->field.tqe_next->field.tqe_prev =	\
      (elm)->field.tqe_prev;	\
```

```
 else       \
  (head)->tqh_last = (elm)->field.tqe_prev;  \
 *(elm)->field.tqe_prev = (elm)->field.tqe_next;   \
 QUEUEDEBUG_TAILQ_POSTREMOVE((elm), field);   \
} while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do {   \
      if (((elm2)->field.tqe_next = (elm)->field.tqe_next) !=  \
    TAILQ_END(head))        \
            (elm2)->field.tqe_next->field.tqe_prev =  \
               &(elm2)->field.tqe_next;    \
      else       \
            (head)->tqh_last = &(elm2)->field.tqe_next;  \
      (elm2)->field.tqe_prev = (elm)->field.tqe_prev;   \
      *(elm2)->field.tqe_prev = (elm2);    \
 QUEUEDEBUG_TAILQ_POSTREMOVE((elm),
 field);   \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do {    \
 if (!TAILQ_EMPTY(head2)) {     \
  *(head1)->tqh_last = (head2)->tqh_first;  \
  (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
  (head1)->tqh_last = (head2)->tqh_last;   \
  TAILQ_INIT((head2));     \
 }       \
} while (/*CONSTCOND*/0)

/*
* Singly-linked Tail queue declarations.
*/
#define STAILQ_HEAD(name, type)      \
struct name {       \
 struct type *stqh_first; /* first element */  \
 struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head)     \
 { NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type)      \
struct {       \
 struct type *stqe_next; /* next element */   \
}

/*
* Singly-linked Tail queue access methods.
*/
```

```
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head)
 == STAILQ_END(head))

/*
* Singly-linked Tail queue functions.
*/
#define STAILQ_INIT(head) do {      \
 (head)->stqh_first = NULL;     \
 (head)->stqh_last = &(head)->stqh_first;    \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do {   \
 if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
  (head)->stqh_last = &(elm)->field.stqe_next;  \
 (head)->stqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do {   \
 (elm)->field.stqe_next = NULL;     \
 *(head)->stqh_last = (elm);     \
 (head)->stqh_last = &(elm)->field.stqe_next;   \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL)\
  (head)->stqh_last = &(elm)->field.stqe_next;  \
 (listelm)->field.stqe_next = (elm);    \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE_HEAD(head, field) do {    \
 if (((head)->stqh_first = (head)->stqh_first->field.stqe_next)
 == NULL) \
  (head)->stqh_last = &(head)->stqh_first;   \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE(head, elm, type, field) do {   \
 if ((head)->stqh_first == (elm)) {    \
  STAILQ_REMOVE_HEAD((head), field);   \
 } else {       \
  struct type *curelm = (head)->stqh_first;  \
  while (curelm->field.stqe_next != (elm))   \
   curelm = curelm->field.stqe_next;  \
  if ((curelm->field.stqe_next =    \
   curelm->field.stqe_next->field.stqe_next) == NULL) \
      (head)->stqh_last = &(curelm)->field.stqe_next; \
```

```
 }       \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->stqh_first);    \
  (var);       \
  (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar)   \
 for ((var) = STAILQ_FIRST((head));    \
     (var) && ((tvar) = STAILQ_NEXT((var), field), 1);  \
     (var) = (tvar))

#define STAILQ_CONCAT(head1, head2) do {    \
 if (!STAILQ_EMPTY((head2))) {     \
  *(head1)->stqh_last
 = (head2)->stqh_first;  \
  (head1)->stqh_last = (head2)->stqh_last;  \
  STAILQ_INIT((head2));     \
 }       \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field)     \
 (STAILQ_EMPTY((head)) ?      \
  NULL :       \
        ((struct type *)(void *)    \
  ((char *)((head)->stqh_last) - offsetof(struct type, field))))


#ifndef _KERNEL
/*
* Circular queue definitions. Do not use. We still keep the macros
* for compatibility but because of pointer aliasing issues their use
* is discouraged!
*/

/*
* __launder_type():  We use this ugly hack to work around the the compiler
* noticing that two types may not alias each other and elide tests in code.
* We hit this in the CIRCLEQ macros when comparing 'struct name *' and
* 'struct type *' (see CIRCLEQ_HEAD()).  Modern compilers (such as GCC
* 4.8) declare these comparisons as always false, causing the code to
* not run as designed.
*
* This hack is only to be used for comparisons and thus can be fully const.
 * Do not use for assignment.
*
* If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
```

```
* this by changing the head/tail sentinal values, but see the note above
* this one.
*/
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
 __asm __volatile("" : "+r" (__x));
 return __x;
}

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)    \
 if ((head)->cqh_first != CIRCLEQ_ENDC(head) &&   \
    (head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
  QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
       __FILE__, __LINE__);    \
 if ((head)->cqh_last != CIRCLEQ_ENDC(head) &&   \
    (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
  QUEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
       __FILE__, __LINE__);
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)   \
 if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) {  \
  if ((head)->cqh_last != (elm))    \
   QUEUEDEBUG_ABORT("CIRCLEQ
 elm last %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 } else {      \
  if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
   QUEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 }       \
 if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) {  \
  if ((head)->cqh_first != (elm))    \
   QUEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 } else {      \
  if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
   QUEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
      (elm), __FILE__, __LINE__);   \
 }
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)   \
 (elm)->field.cqe_next = (void *)1L;    \
 (elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
```

```
#endif

#define CIRCLEQ_HEAD(name, type)     \
struct name {        \
 struct type *cqh_first;  /*
 first element */  \
 struct type *cqh_last;  /* last element */  \
}

#define CIRCLEQ_HEAD_INITIALIZER(head)     \
 { CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type)      \
struct {        \
 struct type *cqe_next;  /* next element */  \
 struct type *cqe_prev;  /* previous element */  \
}

/*
* Circular queue functions.
*/
#define CIRCLEQ_INIT(head) do {      \
 (head)->cqh_first = CIRCLEQ_END(head);    \
 (head)->cqh_last = CIRCLEQ_END(head);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do {  \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field)  \
 (elm)->field.cqe_next = (listelm)->field.cqe_next;  \
 (elm)->field.cqe_prev = (listelm);    \
 if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head))  \
  (head)->cqh_last = (elm);    \
 else        \
  (listelm)->field.cqe_next->field.cqe_prev = (elm); \
 (listelm)->field.cqe_next = (elm);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_BEFORE(head,
 listelm, elm, field) do {  \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field)  \
 (elm)->field.cqe_next = (listelm);    \
 (elm)->field.cqe_prev = (listelm)->field.cqe_prev;  \
 if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head))  \
  (head)->cqh_first = (elm);    \
 else        \
  (listelm)->field.cqe_prev->field.cqe_next = (elm); \
```

```
 (listelm)->field.cqe_prev = (elm);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do {    \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 (elm)->field.cqe_next = (head)->cqh_first;    \
 (elm)->field.cqe_prev = CIRCLEQ_END(head);    \
 if ((head)->cqh_last == CIRCLEQ_ENDC(head))    \
  (head)->cqh_last = (elm);    \
 else        \
  (head)->cqh_first->field.cqe_prev = (elm);  \
 (head)->cqh_first = (elm);     \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do {    \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 (elm)->field.cqe_next
 = CIRCLEQ_END(head);    \
 (elm)->field.cqe_prev = (head)->cqh_last;    \
 if ((head)->cqh_first == CIRCLEQ_ENDC(head))    \
  (head)->cqh_first = (elm);    \
 else        \
  (head)->cqh_last->field.cqe_next = (elm);  \
 (head)->cqh_last = (elm);     \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_REMOVE(head, elm, field) do {    \
 QUEUEDEBUG_CIRCLEQ_HEAD((head), field)    \
 QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field)   \
 if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head))  \
  (head)->cqh_last = (elm)->field.cqe_prev;  \
 else        \
  (elm)->field.cqe_next->field.cqe_prev =   \
      (elm)->field.cqe_prev;    \
 if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head))  \
  (head)->cqh_first = (elm)->field.cqe_next;  \
 else        \
  (elm)->field.cqe_prev->field.cqe_next =   \
      (elm)->field.cqe_next;    \
 QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field)   \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field)    \
 for ((var) = ((head)->cqh_first);    \
  (var) != CIRCLEQ_ENDC(head);    \
  (var)
 = ((var)->field.cqe_next))
```

```
#define CIRCLEQ_FOREACH_REVERSE(var, head, field)   \
 for ((var) = ((head)->cqh_last);    \
  (var) != CIRCLEQ_ENDC(head);    \
  (var) = ((var)->field.cqe_prev))

/*
* Circular queue access methods.
*/
#define CIRCLEQ_FIRST(head)  ((head)->cqh_first)
#define CIRCLEQ_LAST(head)  ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head)  (__launder_type(head))
/* For assignments */
#define CIRCLEQ_END(head)  ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head)      \
   (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field)    \
 (((elm)->field.cqe_next == CIRCLEQ_ENDC(head))   \
     ? ((head)->cqh_first)     \
     : (elm->field.cqe_next))
#define CIRCLEQ_LOOP_PREV(head, elm, field)    \
 (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head))   \
     ? ((head)->cqh_last)     \
     :
 (elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef  __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
```

```
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__ ((__nothrow__))
# define __NTH(fct) __attribute__ ((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct)     fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright"                  # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
 gpep517 build-wheel \
  --wheel-dir .dist \
  --output-fd 3 3>&1 >&2
}

package() {
 gpep517 install-wheel --destdir "$pkgdir" \
  .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
```

```
  flake8-copyright-0.2.4.tar.gz
"
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
 default_prepare

 cargo fetch --target="$CTARGET" --locked
}

build() {
 cargo auditable build --release --frozen
}

check() {
 cargo test --frozen
}

package() {
 install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"
/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
```

```
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
 AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y)      \
 ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) ||   \
  (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

/*
* This file defines data structures for different types of trees:
* splay trees and red-black trees.
*
* A splay tree is a self-organizing
 data structure.  Every operation
* on the tree causes a splay to happen.  The splay moves the requested
* node to the root of the tree and partly rebalances it.
*
* This has the benefit that request locality causes faster lookups as
* the requested nodes move to the top of the tree.  On the other hand,
* every lookup causes memory writes.
*
* The Balance Theorem bounds the total access time for m operations
* and n inserts on an initially empty tree as O((m + n)lg n).  The
```

```
* amortized cost for a sequence of m accesses to a splay tree is O(lg n);
*
* A red-black tree is a binary search tree with the node color as an
* extra attribute.  It fulfills a set of conditions:
* - every search path from the root to a leaf consists of the
*   same number of black nodes,
* - each red node (except for the root) has a black parent,
* - each leaf node is black.
*
* Every operation on a red-black tree is bounded as O(lg n).
* The maximum height of a red-black tree is 2lg (n+1).
 */

#define SPLAY_HEAD(name, type)      \
struct name {       \
 struct type *sph_root; /* root of the tree */   \
}

#define SPLAY_INITIALIZER(root)     \
 { NULL }

#define SPLAY_INIT(root) do {      \
 (root)->sph_root = NULL;     \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type)      \
struct {       \
 struct type *spe_left; /* left element */   \
 struct type *spe_right; /* right element */   \
}

#define SPLAY_LEFT(elm, field)  (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field)  (elm)->field.spe_right
#define SPLAY_ROOT(head)  (head)->sph_root
#define SPLAY_EMPTY(head)  (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do {   \
 SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
 SPLAY_RIGHT(tmp, field) = (head)->sph_root;   \
 (head)->sph_root = tmp;      \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do {   \
 SPLAY_RIGHT((head)->sph_root,
 field) = SPLAY_LEFT(tmp, field); \
 SPLAY_LEFT(tmp, field) = (head)->sph_root;   \
 (head)->sph_root = tmp;      \
```

```
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do {    \
 SPLAY_LEFT(tmp, field) = (head)->sph_root;   \
 tmp = (head)->sph_root;      \
 (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do {    \
 SPLAY_RIGHT(tmp, field) = (head)->sph_root;   \
 tmp = (head)->sph_root;      \
 (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do {  \
 SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
 SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field);\
 SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
 SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
 SPLAY_PROTOTYPE(name, type, field, cmp)    \
void name##_SPLAY(struct name *, struct type *);   \
void name##_SPLAY_MINMAX(struct name *, int);    \
struct type *name##_SPLAY_INSERT(struct name *, struct type *);  \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *);  \
        \
/* Finds the node with the same key as elm */    \
static __inline struct type *      \
name##_SPLAY_FIND(struct name *head, struct type *elm)   \
{        \
 if (SPLAY_EMPTY(head))      \
  return(NULL);      \
 name##_SPLAY(head, elm);     \
 if ((cmp)(elm, (head)->sph_root) == 0)    \
  return (head->sph_root);    \
 return (NULL);       \
}        \
        \
static __inline _sys_tree_h_unused struct type *   \
name##_SPLAY_NEXT(struct name *head, struct type *elm)   \
{        \
 name##_SPLAY(head, elm);     \
 if (SPLAY_RIGHT(elm, field) != NULL) {    \
  elm = SPLAY_RIGHT(elm, field);    \
```

```
  while (SPLAY_LEFT(elm, field) != NULL) {  \
   elm = SPLAY_LEFT(elm, field);   \
  }       \
 }
 else       \
  elm = NULL;      \
 return (elm);       \
}        \
        \
static _sys_tree_h_unused __inline struct type *   \
name##_SPLAY_MIN_MAX(struct name *head, int val)   \
{        \
 name##_SPLAY_MINMAX(head, val);     \
       return (SPLAY_ROOT(head));     \
}

/* Main splay operation.
* Moves node close to the key of elm to top
*/
#define SPLAY_GENERATE(name, type, field, cmp)    \
struct type *       \
name##_SPLAY_INSERT(struct name *head, struct type *elm)  \
{        \
   if (SPLAY_EMPTY(head)) {      \
     SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
   } else {       \
     int __comp;       \
     name##_SPLAY(head, elm);     \
     __comp = (cmp)(elm, (head)->sph_root);   \
     if(__comp < 0) {      \
      SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field);\
      SPLAY_RIGHT(elm, field) = (head)->sph_root;  \
      SPLAY_LEFT((head)->sph_root, field) = NULL;  \
     } else if (__comp > 0) {     \
      SPLAY_RIGHT(elm,
 field) = SPLAY_RIGHT((head)->sph_root, field);\
      SPLAY_LEFT(elm, field) = (head)->sph_root;  \
      SPLAY_RIGHT((head)->sph_root, field) = NULL; \
     } else       \
      return ((head)->sph_root);    \
   }        \
   (head)->sph_root = (elm);      \
   return (NULL);       \
}        \
        \
struct type *       \
name##_SPLAY_REMOVE(struct name *head, struct type *elm)  \
{        \
```

```
 struct type *__tmp;      \
 if (SPLAY_EMPTY(head))      \
  return (NULL);      \
 name##_SPLAY(head, elm);     \
 if ((cmp)(elm, (head)->sph_root) == 0) {   \
  if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
   (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
  } else {      \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
   name##_SPLAY(head, elm);   \
   SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
  }       \
  return (elm);      \
 }        \
 return (NULL);       \
}         \
         \
void         \
name##_SPLAY(struct
 name *head, struct type *elm)   \
{         \
 struct type __node, *__left, *__right, *__tmp;   \
 int __comp;       \
\
 SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
 __left = __right = &__node;     \
\
 while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) {  \
  if (__comp < 0) {     \
   __tmp = SPLAY_LEFT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if ((cmp)(elm, __tmp) < 0){   \
    SPLAY_ROTATE_RIGHT(head, __tmp, field); \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
     break;    \
   }      \
   SPLAY_LINKLEFT(head, __right, field);  \
  } else if (__comp > 0) {    \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if ((cmp)(elm, __tmp) > 0){   \
    SPLAY_ROTATE_LEFT(head, __tmp, field); \
    if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
     break;    \
   }      \
```

```
  SPLAY_LINKRIGHT(head, __left, field);  \
 }       \
 }       \
 SPLAY_ASSEMBLE(head,
 &__node, __left, __right, field);  \
}        \
        \
/* Splay with either the minimum or the maximum element   \
* Used to find minimum or maximum element in tree.   \
*/        \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{        \
 struct type __node, *__left, *__right, *__tmp;   \
\
 SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
 __left = __right = &__node;     \
\
 while (1) {       \
  if (__comp < 0) {     \
   __tmp = SPLAY_LEFT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if (__comp < 0){    \
    SPLAY_ROTATE_RIGHT(head, __tmp, field); \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
     break;    \
   }      \
   SPLAY_LINKLEFT(head, __right, field);  \
  } else if (__comp > 0) {    \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if (__comp > 0) {    \
    SPLAY_ROTATE_LEFT(head, __tmp, field); \
    if (SPLAY_RIGHT((head)->sph_root,
 field) == NULL)\
     break;    \
   }      \
   SPLAY_LINKRIGHT(head, __left, field);  \
  }       \
 }       \
 SPLAY_ASSEMBLE(head, &__node, __left, __right, field);  \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
```

```
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y)  name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y)  name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x)  (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x)  (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head)    \
 for ((x) = SPLAY_MIN(name, head);   \
     (x) != NULL;     \
     (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type)     \
struct name {       \
 struct type *rbh_root; /* root of the tree */  \
}

#define RB_INITIALIZER(root)     \
 {
 NULL }

#define RB_INIT(root) do {      \
 (root)->rbh_root = NULL;    \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED  1
#define RB_ENTRY(type)      \
struct {        \
 struct type *rbe_left;  /* left element */  \
 struct type *rbe_right;  /* right element */  \
 struct type *rbe_parent; /* parent element */  \
 int rbe_color;   /* node color */  \
}

#define RB_LEFT(elm, field)  (elm)->field.rbe_left
#define RB_RIGHT(elm, field)  (elm)->field.rbe_right
#define RB_PARENT(elm, field)  (elm)->field.rbe_parent
#define RB_COLOR(elm, field)  (elm)->field.rbe_color
#define RB_ROOT(head)   (head)->rbh_root
#define RB_EMPTY(head)   (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do {     \
 RB_PARENT(elm, field) = parent;     \
 RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL;  \
 RB_COLOR(elm, field) = RB_RED;     \
```

```
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do {    \
 RB_COLOR(black, field) = RB_BLACK;    \
 RB_COLOR(red, field)
 = RB_RED;     \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do {   \
 (tmp) = RB_RIGHT(elm, field);     \
 if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
  RB_PARENT(RB_LEFT(tmp, field), field) = (elm);  \
 }       \
 RB_AUGMENT(elm);      \
 if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
  if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
   RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
  else      \
   RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
 } else       \
  (head)->rbh_root = (tmp);    \
 RB_LEFT(tmp, field) = (elm);     \
 RB_PARENT(elm, field) = (tmp);     \
 RB_AUGMENT(tmp);      \
 if ((RB_PARENT(tmp, field)))     \
  RB_AUGMENT(RB_PARENT(tmp, field));   \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do {   \
 (tmp) = RB_LEFT(elm, field);     \
 if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field))
 != NULL) { \
  RB_PARENT(RB_RIGHT(tmp, field), field) = (elm);  \
 }       \
 RB_AUGMENT(elm);      \
 if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
  if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
   RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
  else      \
   RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
 } else       \
  (head)->rbh_root = (tmp);    \
 RB_RIGHT(tmp, field) = (elm);     \
 RB_PARENT(elm, field) = (tmp);     \
```

```
 RB_AUGMENT(tmp);      \
 if ((RB_PARENT(tmp, field)))     \
  RB_AUGMENT(RB_PARENT(tmp, field));   \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp)    \
 RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp)   \
 RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr)  \
attr void name##_RB_INSERT_COLOR(struct name *, struct
 type *);  \
attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
attr struct type *name##_RB_INSERT(struct name *, struct type *); \
attr struct type *name##_RB_FIND(struct name *, struct type *);  \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *);   \
attr struct type *name##_RB_PREV(struct type *);   \
attr struct type *name##_RB_MINMAX(struct name *, int);   \
        \

/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp)    \
 RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp)   \
 RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr)  \
attr void        \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm)  \
{        \
 struct type
 *parent, *gparent, *tmp;    \
 while ((parent = RB_PARENT(elm, field)) != NULL &&  \
     RB_COLOR(parent, field) == RB_RED) {   \
  gparent = RB_PARENT(parent, field);   \
  if (parent == RB_LEFT(gparent, field)) {  \
   tmp = RB_RIGHT(gparent, field);   \
   if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent;    \
    continue;    \
   }      \
   if (RB_RIGHT(parent, field) == elm) {  \
```

```
   RB_ROTATE_LEFT(head, parent, tmp, field);\
   tmp = parent;    \
   parent = elm;    \
   elm = tmp;    \
  }      \
  RB_SET_BLACKRED(parent, gparent, field); \
  RB_ROTATE_RIGHT(head, gparent, tmp, field); \
 } else {      \
  tmp = RB_LEFT(gparent, field);   \
  if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
   RB_COLOR(tmp, field) = RB_BLACK; \
   RB_SET_BLACKRED(parent, gparent, field);\
   elm = gparent;    \
   continue;    \
  }      \
  if (RB_LEFT(parent, field) == elm)
 {  \
   RB_ROTATE_RIGHT(head, parent, tmp, field);\
   tmp = parent;    \
   parent = elm;    \
   elm = tmp;    \
  }      \
  RB_SET_BLACKRED(parent, gparent, field); \
  RB_ROTATE_LEFT(head, gparent, tmp, field); \
 }       \
 }       \
 RB_COLOR(head->rbh_root, field) = RB_BLACK;   \
}        \
        \
attr void        \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{        \
 struct type *tmp;      \
 while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
     elm != RB_ROOT(head)) {     \
  if (RB_LEFT(parent, field) == elm) {   \
   tmp = RB_RIGHT(parent, field);   \
   if (RB_COLOR(tmp, field) == RB_RED) {  \
    RB_SET_BLACKRED(tmp, parent, field); \
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    tmp = RB_RIGHT(parent, field);  \
   }      \
   if ((RB_LEFT(tmp, field) == NULL ||  \
       RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
       (RB_RIGHT(tmp, field) == NULL ||  \
       RB_COLOR(RB_RIGHT(tmp,
 field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED;  \
```

```
 elm = parent;    \
 parent = RB_PARENT(elm, field);  \
 } else {     \
 if (RB_RIGHT(tmp, field) == NULL || \
     RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
  struct type *oleft;  \
  if ((oleft = RB_LEFT(tmp, field)) \
      != NULL)   \
   RB_COLOR(oleft, field) = RB_BLACK;\
  RB_COLOR(tmp, field) = RB_RED; \
  RB_ROTATE_RIGHT(head, tmp, oleft, field);\
  tmp = RB_RIGHT(parent, field); \
  }     \
 RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
 RB_COLOR(parent, field) = RB_BLACK; \
 if (RB_RIGHT(tmp, field))  \
  RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
 RB_ROTATE_LEFT(head, parent, tmp, field);\
 elm = RB_ROOT(head);   \
 break;     \
 }      \
} else {      \
 tmp = RB_LEFT(parent, field);   \
 if (RB_COLOR(tmp, field) == RB_RED) {  \
  RB_SET_BLACKRED(tmp, parent, field); \
  RB_ROTATE_RIGHT(head, parent, tmp, field);\
  tmp
= RB_LEFT(parent, field);  \
 }      \
 if ((RB_LEFT(tmp, field) == NULL ||  \
     RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
     (RB_RIGHT(tmp, field) == NULL ||  \
     RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
 RB_COLOR(tmp, field) = RB_RED;  \
 elm = parent;    \
 parent = RB_PARENT(elm, field);  \
 } else {     \
 if (RB_LEFT(tmp, field) == NULL || \
     RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
  struct type *oright;  \
  if ((oright = RB_RIGHT(tmp, field)) \
      != NULL)   \
   RB_COLOR(oright, field) = RB_BLACK;\
  RB_COLOR(tmp, field) = RB_RED; \
  RB_ROTATE_LEFT(head, tmp, oright, field);\
  tmp = RB_LEFT(parent, field); \
  }     \
 RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
```

```
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_LEFT(tmp, field))  \
     RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    elm = RB_ROOT(head);   \
    break;     \
   }      \
  }       \
 }       \
 if
 (elm)      \
  RB_COLOR(elm, field) = RB_BLACK;   \
}        \
        \
attr struct type *       \
name##_RB_REMOVE(struct name *head, struct type *elm)   \
{        \
 struct type *child, *parent, *old = elm;   \
 int color;       \
 if (RB_LEFT(elm, field) == NULL)    \
  child = RB_RIGHT(elm, field);    \
 else if (RB_RIGHT(elm, field) == NULL)    \
  child = RB_LEFT(elm, field);    \
 else {        \
  struct type *left;     \
  elm = RB_RIGHT(elm, field);    \
  while ((left = RB_LEFT(elm, field)) != NULL)  \
   elm = left;     \
  child = RB_RIGHT(elm, field);    \
  parent = RB_PARENT(elm, field);    \
  color = RB_COLOR(elm, field);    \
  if (child)      \
   RB_PARENT(child, field) = parent;  \
  if (parent) {      \
   if (RB_LEFT(parent, field) == elm)  \
    RB_LEFT(parent, field) = child;  \
   else      \
    RB_RIGHT(parent, field) = child; \
   RB_AUGMENT(parent);    \
  } else       \
   RB_ROOT(head)
 = child;    \
  if (RB_PARENT(elm, field) == old)   \
   parent = elm;     \
  (elm)->field = (old)->field;    \
  if (RB_PARENT(old, field)) {    \
   if (RB_LEFT(RB_PARENT(old, field), field) == old)\
    RB_LEFT(RB_PARENT(old, field), field) = elm;\
```

```
  else      \
   RB_RIGHT(RB_PARENT(old, field), field) = elm;\
  RB_AUGMENT(RB_PARENT(old, field));  \
 } else       \
  RB_ROOT(head) = elm;    \
 RB_PARENT(RB_LEFT(old, field), field) = elm;  \
 if (RB_RIGHT(old, field))    \
  RB_PARENT(RB_RIGHT(old, field), field) = elm; \
 if (parent) {      \
  left = parent;     \
  do {      \
   RB_AUGMENT(left);   \
  } while ((left = RB_PARENT(left, field)) != NULL); \
 }       \
 goto color;      \
}        \
parent = RB_PARENT(elm, field);     \
color = RB_COLOR(elm, field);     \
if (child)       \
 RB_PARENT(child, field) = parent;   \
if (parent) {       \
 if (RB_LEFT(parent, field) == elm)   \
  RB_LEFT(parent, field) = child;   \
 else       \
  RB_RIGHT(parent,
field) = child;  \
 RB_AUGMENT(parent);     \
} else        \
 RB_ROOT(head) = child;     \
color:         \
if (color == RB_BLACK)      \
 name##_RB_REMOVE_COLOR(head, parent, child);  \
return (old);       \
}        \
       \
/* Inserts a node into the RB tree */     \
attr struct type *       \
name##_RB_INSERT(struct name *head, struct type *elm)   \
{        \
 struct type *tmp;      \
 struct type *parent = NULL;     \
 int comp = 0;       \
 tmp = RB_ROOT(head);      \
 while (tmp) {       \
  parent = tmp;      \
  comp = (cmp)(elm, parent);    \
  if (comp < 0)      \
   tmp = RB_LEFT(tmp, field);   \
```

```
 else if (comp > 0)     \
  tmp = RB_RIGHT(tmp, field);   \
 else       \
  return (tmp);     \
}       \
RB_SET(elm, parent, field);     \
if (parent != NULL) {      \
 if (comp < 0)      \
  RB_LEFT(parent, field) = elm;   \
 else       \
  RB_RIGHT(parent, field) = elm;   \
 RB_AUGMENT(parent);     \
} else       \
 RB_ROOT(head) = elm;     \
name##_RB_INSERT_COLOR(head,
elm);    \
return (NULL);       \
}       \
       \
/* Finds the node with the same key as elm */    \
attr struct type *       \
name##_RB_FIND(struct name *head, struct type *elm)   \
{       \
 struct type *tmp = RB_ROOT(head);    \
 int comp;       \
 while (tmp) {       \
  comp = cmp(elm, tmp);     \
  if (comp < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else if (comp > 0)     \
   tmp = RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }       \
 return (NULL);       \
}       \
       \
/* Finds the first node greater than or equal to the search key */ \
attr struct type *       \
name##_RB_NFIND(struct name *head, struct type *elm)   \
{       \
 struct type *tmp = RB_ROOT(head);    \
 struct type *res = NULL;     \
 int comp;       \
 while (tmp) {       \
  comp = cmp(elm, tmp);     \
  if (comp < 0) {      \
   res = tmp;     \
```

```
  tmp = RB_LEFT(tmp, field);   \
 }       \
 else if (comp > 0)     \
  tmp =
RB_RIGHT(tmp, field);   \
 else       \
  return (tmp);     \
 }       \
 return (res);       \
}        \
        \
/* ARGSUSED */       \
attr struct type *       \
name##_RB_NEXT(struct type *elm)     \
{        \
 if (RB_RIGHT(elm, field)) {     \
  elm = RB_RIGHT(elm, field);    \
  while (RB_LEFT(elm, field))    \
   elm = RB_LEFT(elm, field);   \
 } else {       \
  if (RB_PARENT(elm, field) &&    \
      (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
   elm = RB_PARENT(elm, field);   \
  else {       \
   while (RB_PARENT(elm, field) &&   \
       (elm == RB_RIGHT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field);  \
   elm = RB_PARENT(elm, field);   \
  }       \
 }       \
 return (elm);       \
}        \
        \
/* ARGSUSED */       \
attr struct type *       \
name##_RB_PREV(struct type *elm)     \
{        \
 if (RB_LEFT(elm, field)) {     \
  elm = RB_LEFT(elm, field);    \
  while (RB_RIGHT(elm, field))    \
   elm = RB_RIGHT(elm, field);   \
 }
 else {       \
  if (RB_PARENT(elm, field) &&    \
      (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
   elm = RB_PARENT(elm, field);   \
  else {       \
   while (RB_PARENT(elm, field) &&   \
```

```
      (elm == RB_LEFT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field);  \
   elm = RB_PARENT(elm, field);   \
  }       \
 }       \
 return (elm);       \
}        \
        \
attr struct type *       \
name##_RB_MINMAX(struct name *head, int val)    \
{        \
 struct type *tmp = RB_ROOT(head);    \
 struct type *parent = NULL;     \
 while (tmp) {       \
  parent = tmp;      \
  if (val < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else       \
   tmp = RB_RIGHT(tmp, field);   \
 }       \
 return (parent);      \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
 y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x)  name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x)  name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head)     \
 for ((x) = RB_MIN(name, head);     \
      (x) != NULL;      \
      (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y)     \
 for ((x) = (y);       \
     ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y)    \
 for ((x) = RB_MIN(name, head);     \
```

```
    ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
     (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head)    \
 for ((x) = RB_MAX(name, head);     \
      (x) != NULL;      \
      (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y)    \
 for ((x) = (y);       \
    ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
     (x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,
 head, y)   \
 for ((x) = RB_MAX(name, head);     \
    ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
     (x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-$_kernver.tar.xz
 revert-broken-uapi.patch
 0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch
```

1.59 readline 8.2.13-r1

1.59.1 Available under license :

```
@c The GNU Free Documentation License.
@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,
@c hence no sectioning command or @node.

@display
Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
@uref{http://fsf.org/}
```

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
@end display

@enumerate 0
@item
PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document @dfn{free} in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative
works of the document
 must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

@item
APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that
contains a notice placed by the copyright holder saying it can be
distributed under the terms of this License. Such a notice grants a
world-wide, royalty-free license, unlimited in duration, to use that
work under the conditions stated herein. The ``Document'', below,
refers
 to any such manual or work. Any member of the public is a
licensee, and is addressed as ``you''. You accept the license if you
copy, modify or distribute the work in a way requiring permission
under copyright law.

A ``Modified Version'' of the Document means any work containing the
Document or a portion of it, either copied verbatim, or with
modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section
of the Document that deals exclusively with the relationship of the
publishers or authors of the Document to the Document's overall
subject (or to related matters) and contains nothing that could fall
directly within that overall subject. (Thus, if the Document is in
part a textbook of mathematics, a Secondary Section may not explain
any mathematics.) The relationship could be a matter of historical
connection with the subject or with related matters, or of legal,
commercial, philosophical, ethical or political
 position regarding
them.

The ``Invariant Sections'' are certain Secondary Sections whose titles
are designated, as being those of Invariant Sections, in the notice
that says that the Document is released under this License. If a
section does not fit the above definition of Secondary then it is not
allowed to be designated as Invariant. The Document may contain zero
Invariant Sections. If the Document does not identify any Invariant
Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed,
as Front-Cover Texts or Back-Cover Texts, in the notice that says that
the Document is released under this License. A Front-Cover Text may
be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent'' copy of the Document means a machine-readable copy,
represented in a format whose specification is available to the
general public, that is suitable for revising the document
straightforwardly with generic text editors or (for
 images composed of
pixels) generic paint programs or (for drawings) some widely available
drawing editor, and that is suitable for input to text formatters or
for automatic translation to a variety of formats suitable for input
to text formatters. A copy made in an otherwise Transparent file
format whose markup, or absence of markup, has been arranged to thwart
or discourage subsequent modification by readers is not Transparent.
An image format is not Transparent if used for any substantial amount
of text. A copy that is not ``Transparent'' is called ``Opaque''.

Examples of suitable formats for Transparent copies include plain
@sc{ascii} without markup, Texinfo input format, La@TeX{} input
format, @acronym{SGML} or @acronym{XML} using a publicly available
@acronym{DTD}, and standard-conforming simple @acronym{HTML},
PostScript or @acronym{PDF} designed for human modification. Examples
of transparent image formats include @acronym{PNG}, @acronym{XCF} and
@acronym{JPG}. Opaque formats
 include proprietary formats that can be
read and edited only by proprietary word processors, @acronym{SGML} or

@acronym{XML} for which the @acronym{DTD} and/or processing tools are
not generally available, and the machine-generated @acronym{HTML},
PostScript or @acronym{PDF} produced by some word processors for
output purposes only.

The ``Title Page'' means, for a printed book, the title page itself,
plus such following pages as are needed to hold, legibly, the material
this License requires to appear in the title page. For works in
formats which do not have any title page as such, ``Title Page'' means
the text near the most prominent appearance of the work's title,
preceding the beginning of the body of the text.

The ``publisher'' means any person or entity that distributes copies
of the Document to the public.

A section ``Entitled XYZ'' means a named subunit of the Document whose
title either is precisely XYZ or contains XYZ in parentheses following
text that translates XYZ in
 another language. (Here XYZ stands for a
specific section name mentioned below, such as ``Acknowledgements'',
``Dedications'', ``Endorsements'', or ``History''.)  To ``Preserve the Title''
of such a section when you modify the Document means that it remains a
section ``Entitled XYZ'' according to this definition.

The Document may include Warranty Disclaimers next to the notice which
states that this License applies to the Document. These Warranty
Disclaimers are considered to be included by reference in this
License, but only as regards disclaiming warranties: any other
implication that these Warranty Disclaimers may have is void and has
no effect on the meaning of this License.

@item
VERBATIM COPYING

You may copy and distribute the Document in any medium, either
commercially or noncommercially, provided that this License, the
copyright notices, and the license notice saying this License applies
to the Document are reproduced in all copies, and that you add no other
conditions whatsoever
 to those of this License. You may not use
technical measures to obstruct or control the reading or further
copying of the copies you make or distribute. However, you may accept
compensation in exchange for copies. If you distribute a large enough
number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and
you may publicly display copies.

@item
COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.
You may add other material on the covers in addition.
Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that
this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item
MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy

of it. In addition, you must do these things in the Modified Version:

@enumerate A
@item
Use in the Title Page (and on the covers, if any) a title distinct
from that of the Document, and from
 those of previous versions
(which should, if there were any, be listed in the History section
of the Document). You may use the same title as a previous version
if the original publisher of that version gives permission.

@item
List on the Title Page, as authors, one or more persons or entities
responsible for authorship of the modifications in the Modified
Version, together with at least five of the principal authors of the
Document (all of its principal authors, if it has fewer than five),
unless they release you from this requirement.

@item
State on the Title page the name of the publisher of the
Modified Version, as the publisher.

@item
Preserve all the copyright notices of the Document.

@item
Add an appropriate copyright notice for your modifications
adjacent to the other copyright notices.

@item
Include, immediately after the copyright notices, a license notice
giving the public permission to use the Modified Version under the
terms of this License, in the form shown in
 the Addendum below.

@item
Preserve in that license notice the full lists of Invariant Sections
and required Cover Texts given in the Document's license notice.

@item
Include an unaltered copy of this License.

@item
Preserve the section Entitled ``History'', Preserve its Title, and add
to it an item stating at least the title, year, new authors, and
publisher of the Modified Version as given on the Title Page. If
there is no section Entitled ``History'' in the Document, create one
stating the title, year, authors, and publisher of the Document as

given on its Title Page, then add an item describing the Modified
Version as stated in the previous sentence.

@item
Preserve the network location, if any, given in the Document for
public access to a Transparent copy of the Document, and likewise
the network locations given in the Document for previous versions
it was based on. These may be placed in the ``History'' section.
You may omit a network location for a work that was published at
least
 four years before the Document itself, or if the original
publisher of the version it refers to gives permission.

@item
For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve
the Title of the section, and preserve in the section all the
substance and tone of each of the contributor acknowledgements and/or
dedications given therein.

@item
Preserve all the Invariant Sections of the Document,
unaltered in their text and in their titles. Section numbers
or the equivalent are not considered part of the section titles.

@item
Delete any section Entitled ``Endorsements''. Such a section
may not be included in the Modified Version.

@item
Do not retitle any existing section to be Entitled ``Endorsements'' or
to conflict in title with any Invariant Section.

@item
Preserve any Warranty Disclaimers.
@end enumerate

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document,
 you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements'', provided it contains
nothing but endorsements of your Modified Version by various
parties---for example, statements of peer review or that the text has
been approved by an organization as the authoritative definition of a

standard.

You may add a passage of up to five words as a Front-Cover Text, and a
passage of up to 25 words as a Back-Cover Text, to the end of the list
of Cover Texts in the Modified Version. Only one passage of
Front-Cover Text and one of Back-Cover Text may be added by (or
through arrangements made by) any one entity. If the Document already
includes a cover text for the same cover, previously added by you or
by arrangement made by the same entity you are acting on behalf
 of,
you may not add another; but you may replace the old one, on explicit
permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License
give permission to use their names for publicity for or to assert or
imply endorsement of any Modified Version.

@item
COMBINING DOCUMENTS

You may combine the Document with other documents released under this
License, under the terms defined in section 4 above for modified
versions, provided that you include in the combination all of the
Invariant Sections of all of the original documents, unmodified, and
list them all as Invariant Sections of your combined work in its
license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and
multiple identical Invariant Sections may be replaced with a single
copy. If there are multiple Invariant Sections with the same name but
different contents, make the title of
 each such section unique by
adding at the end of it, in parentheses, the name of the original
author or publisher of that section if known, or else a unique number.
Make the same adjustment to the section titles in the list of
Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History''
in the various original documents, forming one section Entitled
``History''; likewise combine any sections Entitled ``Acknowledgements'',
and any sections Entitled ``Dedications''. You must delete all
sections Entitled ``Endorsements.''

@item
COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents
released under this License, and replace the individual copies of this

License in the various documents with a single copy that is included in
the collection, provided that you follow the rules of this License for
verbatim copying of each of the documents in all other respects.

You may extract
a single document from such a collection, and distribute
it individually under this License, provided you insert a copy of this
License into the extracted document, and follow this License in all
other respects regarding verbatim copying of that document.

@item
AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate
and independent documents or works, in or on a volume of a storage or
distribution medium, is called an ``aggregate'' if the copyright
resulting from the compilation is not used to limit the legal rights
of the compilation's users beyond what the individual works permit.
When the Document is included in an aggregate, this License does not
apply to the other works in the aggregate which are not themselves
derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these
copies of the Document, then if the Document is less than one half of
the entire aggregate, the Document's Cover Texts may
be placed on
covers that bracket the Document within the aggregate, or the
electronic equivalent of covers if the Document is in electronic form.
Otherwise they must appear on printed covers that bracket the whole
aggregate.

@item
TRANSLATION

Translation is considered a kind of modification, so you may
distribute translations of the Document under the terms of section 4.
Replacing Invariant Sections with translations requires special
permission from their copyright holders, but you may include
translations of some or all Invariant Sections in addition to the
original versions of these Invariant Sections. You may include a
translation of this License, and all the license notices in the
Document, and any Warranty Disclaimers, provided that you also include
the original English version of this License and the original versions
of those notices and disclaimers. In case of a disagreement between
the translation and the original version of this License or a notice
or disclaimer, the original
version will prevail.

If a section in the Document is Entitled ``Acknowledgements'',
``Dedications'', or ``History'', the requirement (section 4) to Preserve
its Title (section 1) will typically require changing the actual
title.

@item
TERMINATION

You may not copy, modify, sublicense, or distribute the Document
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense, or distribute it is void, and
will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license
from a particular copyright holder is reinstated (a) provisionally,
unless and until the copyright holder explicitly and finally
terminates your license, and (b) permanently, if the copyright holder
fails to notify you of the violation by some reasonable means prior to
60 days after the cessation.

Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies
 you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License. If your rights have been terminated and not permanently
reinstated, receipt of a copy of some or all of the same material does
not give you any rights to use it.

@item
FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions
of the GNU Free Documentation License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns. See
@uref{http://www.gnu.org/copyleft/}.

Each version of the License is given a distinguishing version number.
If the Document specifies that
 a particular numbered version of this
License ``or any later version'' applies to it, you have the option of
following the terms and conditions either of that specified version or

of any later version that has been published (not as a draft) by the
Free Software Foundation. If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation. If the Document
specifies that a proxy can decide which future versions of this
License can be used, that proxy's public statement of acceptance of a
version permanently authorizes you to choose that version for the
Document.

@item
RELICENSING

``Massive Multiauthor Collaboration Site'' (or ``MMC Site'') means any
World Wide Web server that publishes copyrightable works and also
provides prominent facilities for anybody to edit those works. A
public wiki that anybody can edit is an example of such a server. A
``Massive Multiauthor Collaboration'' (or ``MMC'')
 contained in the
site means any set of copyrightable works thus published on the MMC
site.

``CC-BY-SA'' means the Creative Commons Attribution-Share Alike 3.0
license published by Creative Commons Corporation, a not-for-profit
corporation with a principal place of business in San Francisco,
California, as well as future copyleft versions of that license
published by that same organization.

``Incorporate'' means to publish or republish a Document, in whole or
in part, as part of another Document.

An MMC is ``eligible for relicensing'' if it is licensed under this
License, and if all works that were first published under this License
somewhere other than this MMC, and subsequently incorporated in whole
or in part into the MMC, (1) had no cover texts or invariant sections,
and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site
under CC-BY-SA on the same site at any time before August 1, 2009,
provided the
 MMC is eligible for relicensing.

@end enumerate

@page
@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and

license notices just after the title page:

```
@smallexample
@group
 Copyright (C)  @var{year}  @var{your name}.
 Permission is granted to copy, distribute and/or modify this document
 under the terms of the GNU Free Documentation License, Version 1.3
 or any later version published by the Free Software Foundation;
 with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
 Texts.  A copy of the license is included in the section entitled ``GNU
 Free Documentation License".
@end group
@end smallexample
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the ``with@dots{}Texts." line with this:

```
@smallexample
@group
   with the Invariant Sections being @var{list their titles}, with
   the Front-Cover
 Texts being @var{list}, and with the Back-Cover Texts
   being @var{list}.
@end group
@end smallexample
```

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.

```
@c Local Variables:
@c ispell-local-pdict: "ispell-dict"
@c End:
```

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed
to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and

modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which
contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies
of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in
whole
or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further

restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution
of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through
any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and
`show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to
freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive

or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees"

and
"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is
not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The
"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that

Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of

your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under
the conditions stated below. Sublicensing is not allowed;
section 10
makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately
publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.

b) The work must carry prominent notices stating that it is
released under this License and any conditions added under section

7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical

medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option

remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating

where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights
granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For
purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting

any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all. For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work. The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you
have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.
Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest

to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where
the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to
do, use the GNU Lesser General
Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.

1.60 ca-certificates-bundle 20250911-r0

1.60.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
*
No license file was found, but licenses were detected in source scan.

-----BEGIN CERTIFICATE-----
MIIH0zCCBbugAwIBAgIIXsO3pkN/pOAwDQYJKoZIhvcNAQEFBQAwQjESMBAGA1UE
AwwJQUNDVlJBSVoxMRAwDgYDVQQLDAdQS0lBQ0NWMQ0wCwYDVQQKDARBQ0NWMQsw
CQYDVQQGEwJFUzAeFw0xMTA1MDUwOTM3MzdaFw0zMDEyMzEwOTM3MzdaMEIxEjAQ
BgNVBAMMCUFDQ1ZSQUlaMTEQMA4GA1UECwwHUEtJQUNDVjENMAsGA1UECgwEQUND
VjELMAkGA1UEBhMCRVMwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCb
qau/YUqXry+XZpp0X9DZlv3P4uRm7x8fRzPCRKPfmt4ftVTdFXxpNRFvu8gMjmoY
HtiP2Ra8EEg2XPBjs5BaXCQ316PWywlxufEBcoSwfdtNgM3802/J+Nq2DoLSRYWo
G2ioPej0RGy9ocLLA76MPhMAhN9KSMDjIgro6TenGEyxCQ0jVn8ETdkXhBilyNpA
lHPrzg5XPAOBOp0KoVdDaaxXbXmQeOW1tDvYvEyNKKGno6e6Ak4l0Squ7a4DIrhr
IA8wKFSVf+DuzgpmndFALW4ir50awQUZ0m/A8p/4e7MCQvtQqR0tkw8jq8bBD5L/
0KIV9VMJcRz/RROE5iZe+OCIHAr8Fraocwa48GOEAqDGWuzndN9wrqODJerWx5eH
k6fGioozl2A3ED6XPm4pFdahD9GILBKfb6qkxkLrQaLjlUPTAYVtjrs78yM2x/47
4KElB0iryYl0/wiPgL/AlmXz7uxLaL2diMMxs0Dx6M/2OLuc5NF/1OVYm3z61PMO
m3WR5LpSLhl+0fXNWhn8ugb2+1KoS5kE3fj5tItQo05iifCHJPqDQsGH+tUtKSpa
cXpkatcnYGMN285J9Y0fkIkyF/hzQ7jSWpOGYdbhdQrqeWZ2iE9x6wQl1gpaepPl
uUsXQA+xtrn13k/c4LOsOxFwYIRKQ26ZIMApcQrAZQIDAQABo4ICyzCCAscwfQYI
KwYBBQUHAQEEcTBvMEwGCCsGAQUFBzAChkBodHRwOi8vd3d3LmFjY3YuZXMvZmls
ZWFkbWluL0FyY2hpdm9zL2NlcnRpZmljYWRvcy9yYWl6YWNjdjEuY3J0MB8GCCsG
AQUFBzABhhNodHRwOi8vb2NzcC5hY2N2LmVzMB0GA1UdDgQWBBTSh7Tj3zcnk1X2
VuqB5TbMjB4/vTAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFNKHtOPfNyeT
VfZW6oHlNsyMHj+9MIIBcwYDVR0gBIIBajCCAWYwggFiBgRVHSAAMIIBWDCCASIG
CCsGAQUFBwICMIIBFB6CARAAQQB1AHQAbwByAGkAZABhAGQAIABkAGUAIABDAGUA
cgB0AGkAZgBpAGMAYQBjAGkA8wBuACAAUgBhAO0AegAgAGQAZQAgAGwAYQAgAEEA
QwBDAFYAIAAoAEEAZwBlAG4AYwBpAGEAIABkAGUAIABUAGUAYwBuAG8AbABvAGcA
7QBhACAAeQAgAEMAZQByAHQAaQBmAGkAYwBhAGMAaQDzAG4AIABFAGwAZQBjAHQA
cgDzAG4AaQBjAGEALAAgAEMASQBGACAAUQA0ADYAMAAxADEANQA2AEUAKQAuACAA
QwBQAFMAIABlAG4AIABoAHQAdABwADoALwAvAHcAdwB3AC4AYQBjAGMAdgAuAGUA
czAwBggrBgEFBQcCARYkaHR0cDovL3d3dy5hY2N2LmVzL2xlZ2lzbGFjaW9uX2Mu
aHRtMFUGA1UdHwROMEwwSqBIoEaGRGh0dHA6Ly93d3cuYWNjdi5lcy9maWxlYWRt
aW4vQXJjaGl2b3MvY2VydGlmaWNhZG9zL3JhaXphY2N2MV9kZXIuY3JsMA4GA1Ud
DwEB/wQEAwIBBjAXBgNVHREEEDAOgQxhY2N2QGFjY3YuZXMwDQYJKoZIhvcNAQEF
BQADggIBAJcxAp/n/UNnSEQU5CmH7UwoZtCPNdpNYbdKl02125DgBS4OxnnQ8pdp
D70ER9m+27Up2pvZrqmZ1dM8MJP1jaGo/AaNRPTKFpV8M9xii6g3+CfYCS0b78gU
JyCpZET/LtZ1qmxNYEAZSUNUY9rizLpm5U9EelvZaoErQNV/+QEnWCzI7UiRfD+m

AM/EKXMRNt6GGT6d7hmKG9Ww7Y49nCrADdg9ZuM8Db3VlFzi4qc1GwQA9j9ajepD
vV+JHanBsMyZ4k0ACtrJJ1vnE5Bc5PUzolVt3OAJTS+xJlsndQAJxGJ3KQhfnlms
tn6tn1QwIgPBHnFk/vk4CpYY3QIUrCPLBhwepH2NDd4nQeit2hW3sCPdK6jT2iWH
7ehVRE2I9DZ+hJp4rPcOVkkO1jMl1oRQQmwgEh0q1b688nCBpHBgvgW1m54ERL5h
I6zppSSMEYCUWqKiuUnSwdzRp+0xESyeGabu4VXhwOrPDYTkF7eifKXeVSUG7szA
h1xA2syVP1XgNce4hL60Xc16gwFy7ofmXx2utYXGJt/mwZrpHgJHnyqobalbz+xF
d3+YJ5oyXSrjhO7FmGYvliAd3djDJ9ew+f7Zfc3Qn48LFFhRny+Lwzgt3uiP1o2H
pPVWQxaZLPSkVrQ0uGE3ycJYgBugl6H8WY3pEfbRD0tVNEYqi4Y7
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFgzCCA2ugAwIBAgIPXZONMGc2yAYdGsdUhGkHMA0GCSqGSIb3DQEBCwUAMDsx
CzAJBgNVBAYTAkVTMREwDwYDVQQKDAhGTk1ULVJDTTEZMBcGA1UECwwQQUMgUkFJ
WiBGTk1ULVJDTTAeFw0wODEwMjkxNTU5NTZaFw0zMDAxMDEwMDAwMDBaMDsxCzAJ
BgNVBAYTAkVTMREwDwYDVQQKDAhGTk1ULVJDTTEZMBcGA1UECwwQQUMgUkFJWiBG
Tk1ULVJDTTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALpxgHpMhm5/
yBNtwMZ9HACXjywMI7sQmkCpGreHiPibVmr75nuOi5KOpyVdWRHbNi63URcfqQgf
BBckWKo3Shjf5TnUV/3XwSyRAZHiItQDwFj8d0fsjz50Q7qsNI1NOHZnjrDIbzAz
WHFctPVrbtQBULgTfmxKo0nRIBnuvMApGGWn3v7v3QqQIecaZ5JCEJhfTzC8PhxF
tBDXaEAUwED653cXeuYLj2VbPNmaUtu1vZ5Gzz3rkQUCwJaydkxNEJY7kvqcfw+Z
374jNUUeAlz+taibmSXaXvMiwzn15Cou08YfxGyqxRxqAQVKL9LFwag0Jl1mpdIC
IfkYtwb1TplvqKtMUejPUBjFd8g5CSxJkjKZqLsXF3mwWsXmo8RZZUc1g16p6DUL
mbvkzSDGm0oGObVo/CK67lWMK07q87Hj/LaZmtVC+nFNCM+HHmpxffnTtOmlcYF7
wk5HlqX2doWjKI/pgG6BU6VtX7hI+cL5NqYuSf+4lsKMB7ObiFj86xsc3i1w4peS
MKGJ47xVqCfWS+2QrYv6YyVZLag13cqXM7zlzced0ezvXg5KkAYmY6252TUtB7p2
ZSysV4999AeU14ECll2jB0nVetBX+RvnU0Z1qrB5QstocQjpYL05ac70r8NWQMet
UqIJ5G+GR4of6ygnXYMgrwTJbFaai0b1AgMBAAGjgYMwgYAwDwYDVR0TAQH/BAUw
AwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFPd9xf3E6Jobd2Sn9R2gzL+H
YJptMD4GA1UdIAQ3MDUwMwYEVR0gADArMCkGCCsGAQUFBwIBFh1odHRwOi8vd3d3
LmNlcnQuZm5tdC5lcy9kcGNzLzANBgkqhkiG9w0BAQsFAAOCAgEAB5BK3/MjTvDD
nFFlm5wiooooMhfNzKWtN/gHiqQxjAb8EZ6WdmF/9ARP67Jpi6Yb+tmLSbkyU+8B1
RXxlDPiyN8+sD8+Nb/kZ94/sHvJwnvDKuO+3/3Y3dlv2bojzr2IyIpMNOmqOFGYM
LVN0V2Ue1bLdI4E7pWYjJ2cJj+F3qkPNZVEI7VFY/uY5+ctHhKQV8Xa7pO6kO8Rf
77IzlhEYt8llvhjho6Tc+hj507wTmzl6NLrTQfv6MooqtyuGC2mDOL7Nii4LcK2N
JpLuHvUBKwrZ1pebbuCoGRw6IYsMHkCtA+fdZn71uSANA+iW+YJF1DngoABd15jm
fZ5nc8OaKveri6E6FO80vFIOiZiaBECEHX5FaZNXzuvO+FB8TxxuBEOb+dY7Ixjp
6o7RTUaN8Tvkasq6+yO3m/qZASlaWFot4/nUbQ4mrcFuNLwy+AwF+mWj2zs3gyLp
1txyM/1d8iC9djwj2ij3+RvrWWTV3F9yfiD8zYm1kGdNYno/Tq0dwzn+evQoFt9B
9kiABdcPUXmsEKvU7ANm5mqwujGSQkBqvjrTcuFqN1W8rB2Vt2lh8kORdOag0wok
RqEIr9baRRmW1FMdW4R58MD3R++Lj8UGrp1MYp3/RgT408m2ECVAdf4WqslKYIYv
uu8wd+RU4riEmViAqhOLUTpPSPaLtrM=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICbjCCAfOgAwIBAgIQYvYybOXE42hcG2LdnC6dlTAKBggqhkjOPQQDAzB4MQsw
CQYDVQQGEwJFUzERMA8GA1UECgwIRk5NVC1SQ00xDjAMBgNVBAsMBUNlcmVzMRgw

FgYDVQRhDA9WQVRFUy1RMjgyNjAwNEoxLDAqBgNVBAMMI0FDIFJBSVogRk5NVC1S
Q00gU0VSVklET1JFUyBTRUdVUk9TMB4XDTE4MTIyMDA5MzczM1oXDTQzMTIyMDA5
MzczM1oweDELMAkGA1UEBhMCRVMxETAPBgNVBAoMCEZOTVQtUkNNMQ4wDAYDVQQL
DAVDZXJlczEYMBYGA1UEYQwPVkFURVMtUTI4MjYwMDRKMSwwKgYDVQQDDCNBQyBS
QUlaIEZOTVQtUkNNIFNFUlZJRE9SRVMgU0VHVVJPUzB2MBAGByqGSM49AgEGBSuB
BAAiA2IABPa6V1PIyqvfNkpSIeSX0oNnnvBlUdBeh8dHsVnyV0ebAAKTRBdp20LH
sbI6GA60XYyzZl2hNPk2LEnb80b8s0RpRBNm/dfF/a82Tc4DTQdxz69qBdKiQ1oK
Um8BA06Oi6NCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYD
VR0OBBYEFAG5L++/EYZg8k/QQW6rcx/n0m5JMAoGCCqGSM49BAMDA2kAMGYCMQCu
SuMrQMN0EfKVrRYj3k4MGuZdpSRea0R7/DjiT8ucRRcRTBQnJlU5dUoDzBOQn5IC
MQD6SmxgiHPz7riYYqnOK8LZiqZwMR2vsJRM60/G49HzYqc8/5MuB1xJAWdpEgJy
v+c=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF7zCCA9egAwIBAgIIDdPjvGz5a7EwDQYJKoZIhvcNAQELBQAwgYQxEjAQBgNV
BAUTCUc2MzI4NzUxMDELMAkGA1UEBhMCRVMxJzAlBgNVBAoTHkFORiBBdXRvcmlk
YWQgZGUgQ2VydGlmaWNhY2lvbjEUMBIGA1UECxMLQU5GIENBIFJhaXoxIjAgBgNV
BAMTGUFORiBTZWN1cmUgU2VydmVyIFJvb3QgQ0EwHhcNMTkwOTA0MTAwMDM4WhcN
MzkwODMwMTAwMDM4WjCBhDESMBAGA1UEBRMJRzYzMjg3NTEwMQswCQYDVQQGEwJF
UzEnMCUGA1UEChMeQU5GIEF1dG9yaWRhZCBkZSBDZXJ0aWZpY2FjaW9uMRQwEgYD
VQQLEwtBTkYgQ0EgUmFpejEiMCAGA1UEAxMZQU5GIFNlY3VyZSBTZXJ2ZXIgUm9v
dCBDQTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBANvrayvmZFSVgpCj
cqQZAZ2cC4Ffc0m6p6zzBE57lgvsEeBbphzOG9INgxwruJ4dfkUyYA8H6XdYfp9q
yGFOtibBTI3/TO80sh9l2Ll49a2pcbnvT1gdpd50IJeh7WhM3pIXS7yr/2WanvtH
2Vdy8wmhrnZEE26cLUQ5vPnHO6RYPUG9tMJJo8gN0pcvB2VSAKduyK9o7PQUlrZX
H1bDOZ8rbeTzPvY1ZNoMHKGESy9LS+IsJJ1tk0DrtSOOMspvRdOoiXsezx76W0OL
zc2oD2rKDF65nkeP8Nm2CgtYZRczuSPkdxl9y0oukntPLxB3sY0vaJxizOBQ+OyR
p1RMVwnVdmPF6GUe7m1qzwmd+nxPrWAI/VaZDxUse6mAq4xhj0oHdkLePfTdsiQz
W7i1o0TJrH93PB0j7IKppuLIBkwC/qxcmZkLLxCKpvR/1Yd0DVlJRfbwcVw5Kda/
SiOL9V8BY9KHcyi1Swr1+KuCLH5zJTIdC2MKF4EA/7Z2Xue0sUDKIbvVgFHlSFJn
LNJhiQcND85Cd8BEc5xEUKDbEAotlRyBr+Qc5RQe8TZBAQIvfXOn3kLMTOmJDVb3
n5HUA8ZsyY/b2BzgQJhdZpmYgG4t/wHFzstGH6wCxkPmrqKEPMVOHj1tyRRM4y5B
u8o5vzY8KhmqQYdOpc5LMnndkEl/AgMBAAGjYzBhMB8GA1UdIwQYMBaAFJxf0Gxj
o1+TypOYCK2Mh6UsXME3MB0GA1UdDgQWBBScX9BsY6Nfk8qTmAitjIelLFzBNzAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOC
AgEATh65isagmD9uw2nAalxJUqzLK114OMHVVISfk/CHGT0sZonrDUL8zPB1hT+L
9IBdeeUXZ701guLyPI59WzbLWoAAKfLOKyzxj6ptBZNscsdW699QIyjlRRA96Gej
rw5VD5AJYu9LWaL2U/HANeQvwSS9eS9OICI7/RogsKQOLHDtdD+4E5UGUcjohybK
pFtqFiGS3XNgnhAY3jyB6ugYw3yJ8otQPr0R4hUDqDZ9MwFsSBXXiJCZBMXM5gf0
vPSQ7RPi6ovDj6MzD8EpTBNO2hVWcXNyglD2mjN8orGoGjR0ZVzO0eurU+AagNjq
OknkJjCb5RyKqKkVMoaZkgoQI1YS4PbOTOK7vtuNknMBZi9iPrJyJ0U27U1W45eZ
/zo1PqVUSlJZS2Db7v54EX9K3BR5YLZrZAPbFYPhor72I5dQ8AkzNqdxliXzuUJ9
2zg/LFis6ELhDtjTO0wugumDLmsx2d1Hhk9tl5EuT+IocTUW0fJz/iUrB0ckYyfI
+PbZa/wSMVYIwFNCr5zQM378BvAxRAMU8Vjq8moNqRGyg77FGr8H6lnco4g175x2
MjxNBiLOFeXdntiP2t7SxDnlF4HPOEfrf4htWRvfn0IUrn7PqLBmZdo3r5+qPeoo
tt7VMVgWglvquxl1AnMaykgaIZOQCo6ThKd9OyMYkomgjaw=

-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFuzCCA6OgAwIBAgIIVwoRl0LE48wwDQYJKoZIhvcNAQELBQAwazELMAkGA1UE
BhMCSVQxDjAMBgNVBAcMBU1pbGFuMSMwIQYDVQQKDBpBY3RhbGlzIFMucC5BLi8w
MzM1ODUyMDk2NzEnMCUGA1UEAwweQWN0YWxpcyBBdXRoZW50aWNhdGlvbiBSb290
IENBMB4XDTExMDkyMjExMjIwMloXDTMwMDkyMjExMjIwMlowazELMAkGA1UEBhMC
SVQxDjAMBgNVBAcMBU1pbGFuMSMwIQYDVQQKDBpBY3RhbGlzIFMucC5BLi8wMzM1
ODUyMDk2NzEnMCUGA1UEAwweQWN0YWxpcyBBdXRoZW50aWNhdGlvbiBSb290IENB
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAp8bEpSmkLO/lGMWwUKNv
UTufClrJwkg4CsIcoBh/kbWHuUA/3R1oHwiD1S0eiKD4j1aPbZkCkpAW1V8IbInX
4ay8IMKx4INRimlNAJZaby/ARH6jDuSRzVju3PvHHkVH3Se5CAGfpiEd9UEtL0z9
KK3giq0itFZljoZUj5NDKd45RnijMCO6zfB9E1fAXdKDa0hMxKufgFpbOr3JpyI/
gCczWw63igxdBzcIy2zSekciRDXFzMwujt0q7bd9Zg1fYVEiVRvjRuPjPdA1Yprb
rxTIW6HMiRvhMCb8oJsfgadHHwTrozmSBp+Z07/T6k9QnBn+locePGX2oxgkg4YQ
51Q+qDp2JE+BIcXjDwL4k5RHILv+1A7TaLndxHqEguNTVHnd25zS8gebLra8Pu2F
be8lEfKXGkJh90qX6IuxEAf6ZYGyojnP9zz/GPvG8VqLWeICrHuS0E4UT1lF9gxe
KF+w6D9Fz8+vm2/7hNN3WpVvrJSEnu68wEqPSpP4RCHiMUVhUE4Q2OM1fEwZtN4F
v6MGn8i1zeQf1xcGDXqVdFUNaBr8EBtiZJ1t4JWgw5QHVw0U5r0F+7if5t+L4sbn
fpb2U8WANFAoWPASUHEXMLrmeGO89LKtmyuy/uE5jF66CyCU3nuDuP/jVo23Eek7
jPKxwV2dpAtMK9myGPW1n0sCAwEAAaNjMGEwHQYDVR0OBBYEFFLYiDrIn3hm7Ynz
ezhwlMkCAjbQMA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUUtiIOsifeGbt
ifN7OHCUyQICNtAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4ICAQAL
e3KHwGCmSUyIWOYdiPcUZEim2FgKDk8TNd81HdTtBjHIgT5q1d07GjLukD0R0i70
jsNjLiNmsGe+b7bAEzlgqqI0JZN1Ut6nna0Oh4lScWoWPBkdg/iaKWW+9D+a2fDz
WochcYBNy+A4mz+7+uAwTc+G02UQGRjRlwKxK3JCaKygvU5a2hi/a5iB0P2avl4V
SM0RFbnAKVy06Ij3Pjaut2L9HmLecHgQHEhb2rykOLpn7VU+Xlff1ANATIGk0k9j
pwlCCRT8AKnCgHNPLsBA2RF7SOp6AsDT6ygBJlh0wcBzIm2Tlf05fbsq4/aC4yyX
X04fkZT6/iyj2HYauE2yOE+b+h1IYHkm4vP9qdCa6HCPSXrW5b0KDtst842/6+Ok
fcvHlXHo2qN8xcL4dJIEG4aspCJTQLas/kx2z/uUMsA1n3Y/buWQbqCmJqK4LL7R
K4X9p2jIugErsWx0Hbhzlefut8cl8ABMALJ+tguLHPPAUJ4lueAI3jZm/zel0btU
ZCzJJ7VLkn5l/9Mt4blOvH+kQSGQQXemOR/qnuOf0GZvBeyqdn6/axag67XH/JJU
LysRJyU3eExRarDzzFhdFPFqSBX/wge2sY0PjlxQRrM9vwGYT7JZVEc+NHt4bVaT
LnPqZih4zR0Uv6CPLy64Lo7yFIrM6bV8+2ydDKXhlg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDTDCCAjSgAwIBAgIId3cGJyapsXwwDQYJKoZIhvcNAQELBQAwRDELMAkGA1UE
BhMCVVMxFDASBgNVBAoMC0FmZmlybVRydXN0MR8wHQYDVQQDDBZBZmZpcm1UcnVz
dCBDb21tZXJjaWFsMB4XDTEwMDEyOTE0MDYwNloXDTMwMTIzMTE0MDYwNlowRDEL
MAkGA1UEBhMCVVMxFDASBgNVBAoMC0FmZmlybVRydXN0MR8wHQYDVQQDDBZBZmZp
cm1UcnVzdCBDb21tZXJjaWFsMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEA9htPZwcroRX1BiLLHwGy43NFBkRJLLtJJRTWzsO3qyxPxkEylFf6EqdbDuKP
Hx6GGaeqtS25Xw2Kwq+FNXkyLbscYjfysVtKPcrNcV/pQr6U6Mje+SJIZMblq8Yr
ba0F8PrVC8+a5fBQpIs7R6UjW3p6+DM/uO+Zl+MgwdYoic+U+7lF7eNAFxHUdPAL
MeIrJmqbTFeurCA+ukV6BfO9m2kVrn1OIGPENXY6BwLJN/3HR+7o8XYdcxXyl6S1

yHp52UKqK39c/s4mT6NmgTWvRLpUHhwwMmWd5jyTXlBOeuM61G7MGvv50jeuJCqr
VwMiKA1JdX+3KNp1v47j3A55MQIDAQABo0IwQDAdBgNVHQ4EFgQUnZPGU4teyq8/
nx4P5ZmVvCT2lI8wDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwDQYJ
KoZIhvcNAQELBQADggEBAFis9AQOzcAN/wr91LoWXym9e2iZWEnStB03TX8nfUYG
XUPGhi4+c7ImfU+TqbbEKpqrIZcUsd6M06uJFdhrJNTxFq7YpFzUf1GO7RgBsZNj
vbz4YYCanrHOQnDiqX0GJX0nof5v7LMeJNrjS1UaADs1tDvZ110w/YETifLCBivt
Z8SOyUOyXGsViQK8YvxO8rUzqrJv0wqiUOP2O+guRMLbZjipM1ZI8W0bM40NjD9g
N53Tym1+NH4Nn3J2ixufcv1SNUFFApYvHLKac0khsUlHRUe072o0EclNmsxZt9YC
nlpOZbWUrhvfKbAW8b8Angc6F2S1BLUjIZkKlTuXfO8=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDTDCCAjSgAwIBAgIIfE8EORzUmS0wDQYJKoZIhvcNAQEFBQAwRDELMAkGA1UE
BhMCVVMxFDASBgNVBAoMC0FmZmlybVRydXN0MR8wHQYDVQQDDBZBZmZpcm1UcnVz
dCBOZXR3b3JraW5nMB4XDTEwMDEyOTE0MDgyNFoXDTMwMTIzMTE0MDgyNFowRDEL
MAkGA1UEBhMCVVMxFDASBgNVBAoMC0FmZmlybVRydXN0MR8wHQYDVQQDDBZBZmZp
cm1UcnVzdCBOZXR3b3JraW5nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEAtITMMxcua5Rsa2FSoOujz3mUTOWUgJnLVWREZY9nZOIG41w3SfYvm4SEHi3y
YJ0wTsyEheIszx6e/jarM3c1RNg1lho9Nuh6DtjVR6FqaYvZ/Ls6rnla1fTWcbua
kCNrmreIdIcMHl+5ni36q1Mr3Lt2PpNMCAiMHqIjHNRqrSK6mQEubWXLviRmVSRL
QESxG9fhwoXA3hA/Pe24/PHxI1Pcv2WXb9n5QHGNfb2V1M6+oF4nI979ptAmDgAp
6zxG8D1gvz9Q0twmQVGeFDdCBKNwV6gbh+0t+nvujArjqWaJGctB+d1ENmHP4ndG
yH329JKBNv3bNPFyfvMMFr20FQIDAQABo0IwQDAdBgNVHQ4EFgQUBx/S55zawm6i
QLSwelAQUHTEyL0wDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwDQYJ
KoZIhvcNAQEFBQADggEBAIlXshZ6qML91tmbmzTCnLQyFE2npN/svqe++EPbkTfO
tDIuUFUaNU52Q3Eg75N3ThVwLofDwR1t3Mu1J9QsVtFSUzpE0nPIxBsFZVpikpzu
QY0x2+c06lkh1QF612S4ZDnNye2v7UsDSKegmQGA3GWjNq5lWUhPgkvIZfFXHeVZ
Lgo/bNjR9eUJtGxUAArgFU2HdW23WJZa3W3SAKD0m0i+wzekujbgfIeFlxoVot4u
olu9rxj5kFDNcFn4J2dHy8egBzp90SxdbBk6ZrV9/ZFvgrG+CJPbFEfxojfHRZ48
x3evZKiT3/Zpg4Jg8klCNO1aAFSFHBY2kgxc+qatv9s=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFRjCCAy6gAwIBAgIIbYwURrGmCu4wDQYJKoZIhvcNAQEMBQAwQTELMAkGA1UE
BhMCVVMxFDASBgNVBAoMC0FmZmlybVRydXN0MRwwGgYDVQQDDBNBZmZpcm1UcnVz
dCBQcmVtaXVtMB4XDTEwMDEyOTE0MTAzNloXDTQwMTIzMTE0MTAzNlowQTELMAkG
A1UEBhMCVVMxFDASBgNVBAoMC0FmZmlybVRydXN0MRwwGgYDVQQDDBNBZmZpcm1U
cnVzdCBQcmVtaXVtMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAxBLf
qV/+Qd3d9Z+K4/as4Tx4mrzY8H96oDMq3I0gW64tb+eT2TZwamjPjlGjhVtnBKAQ
JG9dKILBl1fYSCkTtuG+kU3fhQxTGJoeJKJPj/CihQvL9Cl/0qRY7iZNyaqoe5rZ
+jjeRFcV5fiMyNlI4g0WJx0eyIOFJbe6qlVBzAMiSy2RjYvmia9mx+n/K+k8rNrS
s8PhaJyJ+HoAVt70VZVs+7pk3WKL3wt3MutizCaam7uqYoNMtAZ6MMgpv+0GTZe5
HMQxK9VfvFMSF5yZVylmd2EhMQcuJUmdGPLu8ytxjLW6OQdJd/zvLpKQBY0tL3d7
70O/Nbua2Plzpyzy0FfuKE4mX4+QaAkvuPjcBukumj5Rp9EixAqnOEhss/n/fauG
V+O61oV4d7pD6kh/9ti+I20ev9E2bFhc8e6kGVQa9QPSdubhjL08s9NIS+LI+H+S
qHZGnEJlPqQewQcDWkYtuJfzt9WyVSHvutxMAJf7FJUnM7/oQ0dG0giZFmA7mn7S

5u046uwBHjxIVkkJx0w3AJ6IDsBz4W9m6XJHMD4Q5QsDyZpCAGzFlH5hxIrff4Ia
C1nEWTJ3s7xgaVY5/bQGeyzWZDbZvUjthB9+pSKPKrhC9IK31FOQeE4tGv2Bb0TX
OwF0lkLgAOIua+rF7nKsu7/+6qqo+Nz2snmKtmcCAwEAAaNCMEAwHQYDVR0OBBYE
FJ3AZ6YMItkm9UWrpmVSESfYRaxjMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/
BAQDAgEGMA0GCSqGSIb3DQEBDAUAA4ICAQCzV00QYk465KzquByvMiPIs0laUZx2
KI15qldGF9X1Uva3ROgIRL8YhNILgM3FEv0AVQVhh0HctSSePMTYyPtwni94loMg
Nt58D2kTiKV1NpgIpsbfrM7jWNa3Pt668+s0QNiigfV4Py/VpfzZotReBA4Xrf5B
8OWycvpEgjNC6C1Y91aMYj+6QrCcDFx+LmUmXFNPALJ4fqENmS2NuB2OosSw/WDQ
MKSOyARiqcTtNd56l+0OOF6SL5Nwpamcb6d9Ex1+xghIsV5n61EIJenmJWtSKZGc
0jlzCFfemQa0W50QBuHCAKi4HEoCChTQwUHK+4w1IX2COPKpVJEZNZOUbWo6xbLQ
u4mGk+ibyQ86p3q4ofB4Rvr8Ny/lioTz3/4E2aFooC8k4gmVBtWVyuEklut89pMF
u+1z6S3RdTnX5yTb2E5fQ4+e0BQ5v1VwSJlXMbSc7kqYA5YwH2AG7hsj/oFgIxpH
YoWlzBk0gG+zrBrjn/B7SK3VAdlntqlyk+otZrWyuOQ9PLLvTIzq6we/qzWaVYa8
GKa1qF60g2xraUDTn9zxw2lrueFtCfTxqlB2Cnp9ehehVZZCmTEJ3WARjQUwfuaO
RtGdFNrHF+QFlozEJLUbzxQHskD4o55BhrwE0GuWyCqANP2/7waj3VjFhT0+j/6e
KeC2uAloGRwYQw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIB/jCCAYWgAwIBAgIIdJclisc/elQwCgYIKoZIzj0EAwMwRTELMAkGA1UEBhMC
VVMxFDASBgNVBAoMC0FmZmlybVRydXN0MSAwHgYDVQQDDBdBZmZpcm1UcnVzdCBQ
cmVtaXVtIEVDQzAeFw0xMDAxMjkxNDIwMjRaFw00MDEyMzExNDIwMjRaMEUxCzAJ
BgNVBAYTAlVTMRQwEgYDVQQKDAtBZmZpcm1UcnVzdDEgMB4GA1UEAwwXQWZmaXJt
VHJ1c3QgUHJlbWl1bSBFQ0MwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAAQNMF4bFZ0D
0KF5Nbc6PJJ6yhUczWLznCZcBz3lVPqj1swS6vQUX+iOGasvLkjmrBhDeKzQN8O9
ss0s5kfiGuZjuD0uL3jET9v0D6RoTFVya5UdThhClXjMNzyR4ptlKymjQjBAMB0G
A1UdDgQWBBSaryl6wBE1NSZRMADDav5A1a7WPDAPBgNVHRMBAf8EBTADAQH/MA4G
A1UdDwEB/wQEAwIBBjAKBggqhkjOPQQDAwNnADBkAjAXCfOHiFBar8jAQr9HX/Vs
aobgxCd05DhT1wV/GzTjxi+zygk8N53X57hG8f2h4nECMEJZh0PUUd+60wkyWs6I
flc9nF9Ca/UHLbXwgpP5WW+uZPpY5Yse42O+tYHNbwKMeQ==
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj
ca9HgFB0fW7Y14h29Jlo91ghYPl0hAEvrAIthtOgQ3pOsqTQNroBvo3bSMgHFzZM
9O6II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm
jgSubJrIqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFIQYzIU07LwMlJQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA
A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1Bjjt/msv0tadQ1wUs

N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv
o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU
5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rqXRfboQnoZsG4q5WTP468SQvvG5
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFQTCCAymgAwIBAgITBmyf0pY1hp8KD+WGePhbJruKNzANBgkqhkiG9w0BAQwF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6
b24gUm9vdCBDQSAyMB4XDTE1MDUyNjAwMDAwMFoXDTQwMDUyNjAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMjCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK2Wny2cSkxK
gXlRmeyKy2tgURO8TW0G/LAIjd0ZEGrHJgw12MBvIITplLGbhQPDW9tK6Mj4kHbZ
W0/jTOgGNk3Mmqw9DJArktQGGWCsN0R5hYGCrVo34A3MnaZMUnbqQ523BNFQ9lXg
1dKmSYXpN+nKfq5clU1Imj+uIFptiJXZNLhSGkOQsL9sBbm2eLfq0OQ6PBJTYv9K
8nu+NQWpEjTj82R0Yiw9AElaKP4yRLuH3WUnAnE72kr3H9rN9yFVkE8P7K6C4Z9r
2UXTu/Bfh+08LDmG2j/e7HJV63mjrdvdfLC6HM783k81ds8P+HgfajZRRidhW+me
z/CiVX18JYpvL7TFz4QuK/0NURBs+18bvBt+xa47mAExkv8LV/SasrlX6avvDXbR
8O70zoan4G7ptGmh32n2M8ZpLpcTnqWHsFcQgTfJU7O7f/aS0ZzQGPSSbtqDT6Zj
mUyl+17vIWR6IF9sZIUVyzfpYgwLKhbcAS4y2j5L9Z469hdAlO+ekQiG+r5jqFoz
7Mt0Q5X5bGlSNscpb/xVA1wf+5+9R+vnSUeVC06JIglJ4PVhHvG/LopyboBZ/1c6
+XUyo05f7O0oYtlNc/LMgRdg7c3r3NunysV+Ar3yVAhU/bQtCSwXVEqY0VThUWcI
0u1ufm8/0i2BWSlmy5A5lREedCf+3euvAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMB
Af8wDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQWBBSwDPBMMPQFWAJI/TPlUq9LhONm
UjANBgkqhkiG9w0BAQwFAAOCAgEAqqiAjw54o+Ci1M3m9Zh6O+oAA7CXDpO8Wqj2
LIxyh6mx/H9z/WNxeKWHWc8w4Q0QshNabYL1auaAn6AFC2jkR2vHat+2/XcycuUY
+gn0oJMsXdKMdYV2ZZAMA3m3MSNjrXiDCYZohMr/+c8mmpJ5581LxedhpxfL86kS
k5Nrp+gvU5LEYFiwzAJRGFuFjWJZY7attN6a+yb3ACfAXVU3dJnJUH/jWS5E4ywl
7uxMMne0nxrpS10gxdr9HIcWxkPo1LsmmkVwXqkLN1PiRnsn/eBG8om3zEK2yygm
btmlyTrIQRNg91CMFa6ybRoVGld45pIq2WWQgj9sAq+uEjonljYE1x2igGOpm/Hl
urR8FLBOybEfdF849lHqm/osohHUqS0nGkWxr7JOcQ3AWEbWaQbLU8uz/mtBzUF+
fUwPfHJ5elnNXkoOrJupmHN5fLT0zLm4BwyydFy4x2+IoZCn9Kr5v2c69BoVYh63
n749sSmvZ6ES8lgQGVMDMBu4Gon2nL2XA46jCfMdiyHxtN/kHNGfZQIG6lzWE7OE
76KlXIx3KadowGuuQNKotOrN8I1LOJwZmhsoVLiJkO/KdYE+HvJkJMcYr07/R54H
9jVlpNMKVv/1F2Rs76giJUmTtt8AF9pYfl3uxRuw0dFfIRDH+fO6AgonB8Xx1sfT
4PsJYGw=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIBtjCCAVugAwIBAgITBmyf1XSXNmY/Owua2eiedgPySjAKBggqhkjOPQQDAjA5
MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6b24g
Um9vdCBDQSAzMB4XDTE1MDUyNjAwMDAwMFoXDTQwMDUyNjAwMDAwMFowOTELMAkG
A1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJvb3Qg
Q0EgMzBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABCmXp8ZBf8ANm+gBG1bG8lKl
ui2yEujSLtf6ycXYqm0fc4E7O5hrOXwzpcVOho6AF2hiRVd9RFgdszflZwjrZt6j
QjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQWBBSr

ttvXBp43rDCGB5Fwx5zEGbF4wDAKBggqhkjOPQQDAgNJADBGAiEA4IWSoxe3jfkr
BqWTrBqYaGFy+uGh0PsceGCmQ5nFuMQCIQCcAu/xlJyzlvnrxir4tiz+OpAUFteM
YyRIHN8wfdVoOw==
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIB8jCCAXigAwIBAgITBmyf18G7EEwpQ+Vxe3ssyBrBDjAKBggqhkjOPQQDAzA5
MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6b24g
Um9vdCBDQSA0MB4XDTE1MDUyNjAwMDAwMFoXDTQwMDUyNjAwMDAwMFowOTELMAkG
A1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJvb3Qg
Q0EgNDB2MBAGByqGSM49AgEGBSuBBAAiA2IABNKrijdPo1MN/sGKe0uoe0ZLY7Bi
9i0b2whxIdIA6GO9mif78DluXeo9pcmBqqNbIJhFXRbb/egQbeOc4OO9X4Ri83Bk
M6DLJC9wuoihKqB1+IGuYgbEgds5bimwHvouXKNCMEAwDwYDVR0TAQH/BAUwAwEB
/zAOBgNVHQ8BAf8EBAMCAYYwHQYDVR0OBBYEFNPsxzplbszh2naaVvuc84ZtV+WB
MAoGCCqGSM49BAMDA2gAMGUCMDqLIfG9fhGt0O9Yli/W651+kI0rz2ZVwyzjKKlw
CkcO8DdZEv8tmZQoTipPNU0zWgIxAOp1AE47xDqUEpHJWEadIRNyp4iciuRMStuW
1KyLa2tJElMzrdfkviT8tQp21KW8EA==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDdzCCAl+gAwIBAgIIXDPLYixfszIwDQYJKoZIhvcNAQELBQAwPDEeMBwGA1UE
AwwVQXRvcyBUcnVzdGVkUm9vdCAyMDExMQ0wCwYDVQQKDARBdG9zMQswCQYDVQQG
EwJERTAeFw0xMTA3MDcxNDU4MzBaFw0zMDEyMzEyMzU5NTlaMDwxHjAcBgNVBAMM
FUF0b3MgVHJ1c3RlZFJvb3QgMjAxMTENMAsGA1UECgwEQXRvczELMAkGA1UEBhMC
REUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCVhTuXbyo7LjvPpvMp
Nb7PGKw+qtn4TaA+Gke5vJrf8v7MPkfoepbCJI419KkM/IL9bcFyYie96mvr54rM
VD6QUM+A1JX76LWC1BTFtqlVJVfbsVD2sGBkWXppzwO3bw2+yj5vdHLqqjAqc2K+
SZFhyBH+DgMq92og3AIVDV4VavzjgsG1xZ1kCWyjWZgHJ8cblithdHFsQ/H3NYkQ
4J7sVaE3IqKHBAUsR320HLliKWYoyrfhk/WklAOZuXCFteZI6o1Q/NnezG8HDt0L
cp2AMBYHlT8oDv3FdU9T1nSatCQujgKRz3bFmx5VdJx4IbHwLfELn8LVlhgf8FQi
eowHAgMBAAGjfTB7MB0GA1UdDgQWBBSnpQaxLKYJYO7Rl+lwrrw7GWzbITAPBgNV
HRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFKelBrEspglg7tGX6XCuvDsZbNshMBgG
A1UdIAQRMA8wDQYLKwYBBAGwLQMEAQEwDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3
DQEBCwUAA4IBAQAmdzTblEiGKkGdLD4GkGDEjKwLVLgfuXvTBznk+j57sj1O7Z8j
vZfza1zv7v1Apt+hk6EKhqzvINB5Ab149xnYJDE0BAGmuhWawyfc2E8PzBhj/5kP
DpFrdRbhIfzYJsdHt6bPWHJxfrrhTZVHO8mvbaG0weyJ9rQPOLXiZNwlz6bb65pc
maHFCN795trV1lpFDMS3wrUU77QR/w4VtfX128a961qn8FYiqTxlVMYVqL2Gns2D
lmh6cYGJ4Qvh6hEbaAjMaZ7snkGeRDImeuKHCnE96+RapNLbxc3G3mB/ufNPRJLv
KrcYPqcZ2Qt9sTdBQrC6YB3y/gkRsPCHe6ed
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICFTCCAZugAwIBAgIQPZg7pmY9kGP3fiZXOATvADAKBggqhkjOPQQDAzBMMS4w
LAYDVQQDDCVBdG9zIFRydXN0ZWRSb290IFJvb3QgQ0EgRUNDIFRMUyAyMDIxMQ0w
CwYDVQQKDARBdG9zMQswCQYDVQQGEwJERTAeFw0yMTA0MjIwOTI2MjNaFw00MTA0
MTcwOTI2MjJaMEwxLjAsBgNVBAMMJUF0b3MgVHJ1c3RlZFJvb3QgUm9vdCBDQSBF

Q0MgVExTIDIwMjExDTALBgNVBAoMBEF0b3MxCzAJBgNVBAYTAkRFMHYwEAYHKoZI
zj0CAQYFK4EEACIDYgAEloZYKDcKZ9Cg3iQZGeHkBQcfl+3oZIK59sRxUM6KDP/X
tXa7oWyTbIOiaG6l2b4siJVBzV3dscqDY4PMwL502eCdpO5KTlbgmClBk1IQ1SQ4
AjJn8ZQSb+/Xxd4u/RmAo0IwQDAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBR2
KCXWfeBmmnoJsmo7jjPXNtNPojAOBgNVHQ8BAf8EBAMCAYYwCgYIKoZIzj0EAwMD
aAAwZQIwW5kp85wxtolrbNa9d+F851F+uDrNozZffPc8dz7kUK2o59JZDCaOMDtu
CCrCp1rIAjEAmeMM56PDr9NJLkaCI2ZdyQAUEv049OGYa3cpetskz2VAv9LcjBHo
9H1/IISpQuQo
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFZDCCA0ygAwIBAgIQU9XP5hmTC/srBRLYwiqipDANBgkqhkiG9w0BAQwFADBM
MS4wLAYDVQQDDCVBdG9zIFRydXN0ZWRSb290IFJvb3QgQ0EgUlNBIFRMUyAyMDIx
MQ0wCwYDVQQKDARBdG9zMQswCQYDVQQGEwJERTAeFw0yMTA0MjIwOTIxMTBaFw00
MTA0MTcwOTIxMDlaMEwxLjAsBgNVBAMMJUF0b3MgVHJ1c3RlZFJvb3QgUm9vdCBD
QSBSU0EgVExTIDIwMjExDTALBgNVBAoMBEF0b3MxCzAJBgNVBAYTAkRFMIICIjAN
BgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAtoAOxHm9BYx9sKOdTSJNy/BBl01Z
4NH+VoyX8te9j2y3I49f1cTYQcvyAh5x5en2XssIKl4w8i1mx4QbZFc4nXUtVsYv
Ye+W/CBGvevUez8/fEc4BKkbqlLfEzfTFRVOvV98r61jx3ncCHvVoOX3W3WsgFWZ
kmGbzSoXfduP9LVq6hdKZChmFSlsAvFr1bqjM9xaZ6cF4r9lthawEO3NUDPJcFDs
GY6wx/J0W2tExn2WuZgIWWbeKQGb9Cpt0xU6kGpn8bRrZtkh68rZYnxGEFzedUln
nkL5/nWpo63/dgpnQOPF943HhZpZnmKaau1Fh5hnstVKPNe0OwANwI8f4UDErmwh
3El+fsqyjW22v5MvoVw+j8rtgI5Y4dtXz4U2OLJxpAmMkokIiEjxQGMYsluMWuPD
0xeqqxmjLBvk1cbiZnrXghmmOxYsL3GHX0WelXOTwkKBIROW1527k2gV+p2kHYzy
geBYBr3JtuP2iV2J+axEoctr+hbxx1A9JNr3w+SH1VbxT5Aw+kUJWdo0zuATHAR8
ANSbhqRAvNncTFd+rrcztl524WWLZt+NyteYr842mIycg5kDcPOvdO3GDjbnvezB
c6eUWsuSZIKmAMFwoW4sKeFYV+xafJlrJaSQOoD0IJ2azsct+bJLKZWD6TWNp0lI
pw9MGZHQ9b8Q4HECAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
dEmZ0f+0emhFdcN+tNzMzjkz2ggwDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3DQEB
DAUAA4ICAQAjQ1MkYlxt/T7Cz1UAbMVWiLkO3TriJQ2VSpfKgInuKs1l+NsW4AmS
4BjHeJi78+xCUvuppILXTdiK/ORO/auQxDh1MoSf/7OwKwIzNsAQkG8dnK/haZPs
o0UvFJ/1TCplQ3IM98P4lYsU84UgYt1UU90s3BiVaU+DR3BAM1h3Egyi61IxHkzJ
qM7F78PRreBrAwA0JrRUITWXAdxfG/F851X6LWh3e9NpzNMOa7pNdkTWwhWaJuyw
xfW70Xp0wmzNxbVe9kzmWy2B27O3Opee7c9GslA9hGCZcbUztVdF5kJHdWoOsAgM
rr3e97sPWD2PAzHoPYJQyi9eDF20l74gNAf0xBLh7tew2VktafcxBPTy+av5EzH4
AXcOPUIjJsyacmdRIXrMPIWo6iFqO9taPKU0nprALN+AnCng33eU0aKAQv9qTFsR
0PXNor6uzFFcw9VUewyu1rkGd4Di7wcaaMxZUa1+XGdrudviB0JbuAEFWDlN5LuY
o7Ey7Nmj1m+UI/87tyll5gfp77YZ6ufCOB0yiJA8EytuzO+rdwY0d4RPcuSBhPm5
dDTedk+SKlOxJTnbPP/lPqYO5Wue/9vsL3SD3460s6neFE3/MaNFcyT6lSnMEpcE
oji2jbDwN/zIIX8/syQbPYtuzE2wFg2WHYMfRsCbvUOZ58SWLs5fyQ==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIGFDCCA/ygAwIBAgIIG3Dp0v+ubHEwDQYJKoZIhvcNAQELBQAwUTELMAkGA1UE
BhMCRVMxQjBABgNVBAMMOUF1dG9yaWRhZCBkZSBDZXJ0aWZpY2FjaW9uIEZpcm1h
cHJvZmVzaW9uYWwgQ0lGIEE2MjYzNDA2ODAeFw0xNDA5MjMxNTIyMDdaFw0zNjA1

MDUxNTIyMDdaMFExCzAJBgNVBAYTAkVTMUIwQAYDVQQDDDlBdXRvcmlkYWQgZGUg
Q2VydGlmaWNhY2lvbiBGaXJtYXByb2Zlc2lvbmFsIENJRiBBNjI2MzQwNjgwggIi
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDKlmuO6vj78aI14H9M2uDDUtd9
thDIAl6zQyrET2qyyhxdKJp4ERppWVevtSBC5IsP5t9bpgOSL/UR5GLXMnE42QQM
cas9UX4PB99jBVzpv5RvwSmCwLTaUbDBPLutN0pcyvFLNg4kq7/DhHf9qFD0sefG
L9ItWY16Ck6WaVICqjaY7Pz6FIMMNx/Jkjd/14Et5cS54D40/mf0PmbR0/RAz15i
NA9wBj4gGFrO93IbJWyTdBSTo3OxDqqHECNZXyAFGUftaI6SEspd/NYrspI8IM/h
X68gvqB2f3bl7BqGYTM+53u0P6APjqK5am+5hyZvQWyIplD9amML9ZMWGxmPsu2b
m8mQ9QEM3xk9Dz44I8kvjwzRAv4bVdZO0I08r0+k8/6vKtMFnXkIoctXMbScyJCy
Z/QYFpM6/EfY0XiWMR+6KwxfXZmtY4laJCB22N/9q06mIqqdXuYnin1oKaPnirja
EbsXLZmdEyRG98Xi2J+Of8ePdG1asuhy9azuJBCtLxTa/y2aRnFHvkLfuwHb9H/T
KI8xWVvTyQKmtFLKbpf7Q8UIJm+K9Lv9nyiqDdVF8xM6HdjAeI9BZzwelGSuewvF
6NkBiDkal4ZkQdU7hwxu+g/GvUgUvzlN1J5Bto+WHWOWk9mVBngxaJ43BjuAiUVh
OSPHG0SjFeUc+JIwuwIDAQABo4HvMIHsMB0GA1UdDgQWBBRlzeurNR4APn7VdMAc
tHNHDhpkLzASBgNVHRMBAf8ECDAGAQH/AgEBMIGmBgNVHSAEgZ4wgZswgZgGBFUd
IAAwgY8wLwYIKwYBBQUHAgEWI2h0dHA6Ly93d3cuZmlybWFwcm9mZXNpb25hbC5j
b20vY3BzMFwGCCsGAQUFBwICMFAeTgBQAGEAcwBlAG8AIABkAGUAIABsAGEAIABC
AG8AbgBhAG4AbwB2AGEAIAA0ADcAIABCAGEAcgBjAGUAbABvAG4AYQAgADAAOAAw
ADEANzAOBgNVHQ8BAf8EBAMCAQYwDQYJKoZIhvcNAQELBQADggIBAHSHKAIrdx9m
iWTtj3QuRhy7qPj4Cx2Dtjqn6EWKB7fgPiDL4QjbEwj4KKE1soCzC1HA01aajTNF
Sa9J8OA9B3pFE1r/yJfY0xgsfZb43aJlQ3CTkBW6kN/oGbDbLIpgD7dvlAceHabJ
hfa9NPhAeGIQcDq+fUs5gakQ1JZBu/hfHAsdCPKxsIl68veg4MSPi3i1O1ilI45P
Vf42O+AMt8oqMEEgtIDNrvx2ZnOorm7hfNoD6JQg5iKj0B+QXSBTFCZX2lSX3xZE
EAEeiGaPcjiT3SC3NL7X8e5jjkd5KAb881lFJWAiMxujX6i6KtoaPc1A6ozuBRWV
1aUsIC+nmCjuRfzxuIgALI9C2lHVnOUTaHFFQ4ueCyE8S1wF3BqfmI7avSKecs2t
CsvMo2ebKHTEm9caPARYpoKdrcd7b/+Alun4jWq9GJAd/0kakFI3ky88Al2CdgtR
5xbHV/g4+afNmyJU72OwFW1TZQNKXkqgsqeOSQBZONXH9IBk9W6VULgRfhVwOEqw
f9DEMnDAGf/JOC0ULGb0QkTmVXYbgBVX/8Cnp6o5qtjTcNAuuuuUavpfNIbnYrX9
ivAwhZTJryQCL2/W3Wf+47BVTwSYT6RBVuKT0Gro1vP7ZeDOdcQxWQzugsgMYDNK
GbqEZycPvEJdvSRUDewdcAZfpLz6IHxV
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFdDCCA1ygAwIBAgIQVW9l47TZkGobCdFsPsBsIDANBgkqhkiG9w0BAQsFADBU
MQswCQYDVQQGEwJDTjEmMCQGA1UECgwdQkVJSklORyBDRVJUSUZJQ0FURSBBVVRI
T1JJVFkxHTAbBgNVBAMMFEJKQ0EgR2xvYmFsIFJvb3QgQ0ExMB4XDTE5MTIxOTAz
MTYxN1oXDTQ0MTIxMjAzMTYxN1owVDELMAkGA1UEBhMCQ04xJjAkBgNVBAoMHUJF
SUpJTkcgQ0VSVElGSUNBVEUgQVVUSE9SSVRZMR0wGwYDVQQDDBRCSkNBIEdsb2Jh
bCBSb290IENBMTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAPFmCL3Z
xRVhy4QEQaVpN3cdwbB7+sN3SJATcmTRuHyQNZ0YeYjjlwE8R4HyDqKYDZ4/N+AZ
spDyRhySsTphzvq3Rp4Dhtczbu33RYx2N95ulpH3134rhxfVizXuhJFyV9xgw8O5
58dnJCNPYwpj9mZ9S1WnP3hkSWkSl+BMDdMJoDIwOvqfwPKcxRIqLhy1BDPapDgR
at7GGPZHOiJBhyL8xIkoVNiMpTAK+BcWyqw3/XmnkRd4OJmtWO2y3syJfQOcs4ll
5+M7sSKGjwZteAf9kRJ/sGsciQ35uMt0WwfCyPQ10WRjeulumijWML3mG90Vr4Tq
nMfK9Q7q8l0ph49pczm+LiRvRSGsxdRpJQaDrXpIhRMsDQa4bHlW/KNnMoH1V6XK
V0Jp6VwkYe/iMBhORJhVb3rCk9gZtt58R4oRTklH2yiUAguUSiz5EtBP6DF+bHq/
pj+bOT0CFqMYs2esWz8sgytnOYFcuX6U1WTdno9uruh8W7TXakdI136z1C2OVnZO

z2nxbkRs1CTqjSShGL+9V/6pmTW12xB3uD1IutbB5/EjPtffhZ0nPNRAvQoMvfXn
jSXWgXSHRtQpdaJCbPdzied9v3pKH9MiyRVVz99vfFXQpIsHETdfg6YmV6YBW37+
WGgHqel62bno/1Afq8K0wM7o6v0PvY1NuLxxAgMBAAGjQjBAMB0GA1UdDgQWBBTF
7+3M2I0hxkjk49cULqcWk+WYATAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQE
AwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAUoKsITQfI/Ki2Pm4rzc2IInRNwPWaZ+4
YRC6ojGYWUfo0Q0lHhVBDOAqVdVXUsv45Mdpox1NcQJeXyFFYEhcCY5JEMEE3Kli
awLwQ8hOnThJdMkycFRtwUf8jrQ2ntScvd0g1lPJGKm1Vrl2i5VnZu69mP6u775u
+2D2/VnGKhs/I0qUJDAnyIm860Qkmss9vk/Ves6OF8tiwdneHg56/0OGNFK8YT88
X7vZdrRTvJez/opMEi4r89fO4aL/3Xtw+zuhTaRjAv04l5U/BXCga99igUOLtFkN
SoxUnMW7gZ/NfaXvCyUeOiDbHPwfmGcCCtRzRBPbUYQaVQNW4AB+dAb/OMRyHdOo
P2gxXdMJxy6MW2Pg6Nwe0uxhHvLe5e/2mXZgLR6UcnHGCyoyx5JO1UbXHfmpGQrI
+pXObSOYqgs4rZpWDW+N8TEAiMEXnM0ZNjX+VVOg4DwzX5Ze4jLp3zO7Bkqp2IRz
znfSxqxx4VyjHQy7Ct9f4qNx2No3WqB4K/TUfet27fJhcKVlmtOJNBir+3I+17Q9
eVzYH6Eze9mCUAyTF6ps3MKCuwJXNq+YJyo5UOGwifUll35HaBC07HPKs5fRJNz2
YqAo07WjuGS3iGJCz51TzZm+ZGiPTx4SSPfSKcOYKMryMguTjClPPGAyzQWWYezy
r/6zcCwupvI=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICJTCCAaugAwIBAgIQLBcIfWQqwP6FGFkGz7RK6zAKBggqhkjOPQQDAzBUMQsw
CQYDVQQGEwJDTjEmMCQGA1UECgwdQkVJSklORyBDRVJUSUZJQ0FURSBBVVRIT1JJ
VFkxHTAbBgNVBAMMFEJKQ0EgR2xvYmFsIFJvb3QgQ0EyMB4XDTE5MTIxOTAzMTgy
MVoXDTQ0MTIxMjAzMTgyMVowVDELMAkGA1UEBhMCQ04xJjAkBgNVBAoMHUJFSUpJ
TkcgQ0VSVElGSUNBVEUgQVVUSE9SSVRZMR0wGwYDVQQDDBRCSkNBIEdsb2JhbCBS
b290IENBMjB2MBAGByqGSM49AgEGBSuBBAAiA2IABJ3LgJGNU2e1uVCxA/jlSR9B
IgmwUVJY1is0j8USRhTFiy8shP8sbqjV8QnjAyEUxEM9fMEsxEtqSs3ph+B99iK+
+kpRuDCK/eHeGBIK9ke35xe/J4rUQUyWPGCWwf0VHKNCMEAwHQYDVR0OBBYEFNJK
sVF/BvDRgh9Obl+rg/xI1LCRMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQD
AgEGMAoGCCqGSM49BAMDA2gAMGUCMBq8W9f+qdJUDkpd0m2xQNz0Q9XSSpkZElaA
94M04TVOSG0ED1cxMDAtsaqdAzjbBgIxAMvMh1PLet8gUXOQwKhbYdDFUDn9hf7B
43j4ptZLvZuHjw/l1lOWqzzIQNph91Oj9w==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFWTCCA0GgAwIBAgIBAjANBgkqhkiG9w0BAQsFADBOMQswCQYDVQQGEwJOTzEd
MBsGA1UECgwUQnV5cGFzcyBBUy05ODMxNjMzMjcxIDAeBgNVBAMMF0J1eXBhc3Mg
Q2xhc3MgMiBSb290IENBMB4XDTEwMTAyNjA4MzgwM1oXDTQwMTAyNjA4MzgwM1ow
TjELMAkGA1UEBhMCTk8xHTAbBgNVBAoMFEJ1eXBhc3MgQVMtOTgzMTYzMzI3MSAw
HgYDVQQDDBdCdXlwYXNzIENsYXNzIDIgUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEB
BQADggIPADCCAgoCggIBANfHXvfBB9R3+0Mh9PT1aeTuMgHbo4Yf5FkNuud1g1Lr
6hxhFUi7HQfKjK6w3Jad6sNgkoaCKHOcVgb/S2TwDCo3SbXlzwx87vFKu3MwZfPV
L4O2fuPn9Z6rYPnT8Z2SdIrkHJasW4DptfQxh6NR/Md+oW+OU3fUl8FVM5I+GC91
1K2GScuVr1QGbNgGE41b/+EmGVnAJLqBcXmQRFBoJJRfuLMR8SlBYaNByyM21cHx
MlAQTn/0hpPshNOOvEu/XAFOBz3cFIqUCqTqc/sLUegTBxj6DvEr0VQVfTzh97QZ
QmdiXnfgolXsttlpF9U6r0TtSsWe5HonfOV116rLJeffawrbD02TTqigzXsu8lkB
arcNuAeBfos4GzjmCleZPe4h6KP1DBbdi+w0jpwqHAAVF41og9JwnxgIzRFo1clr

Us3ERo/ctfPYV3Me6ZQ5BL/T3jjetFPsaRyifsSP5BtwrfKi+fv3FmRmaZ9JUaLi
FRhnBkp/1Wy1TbMz4GHrXb7pmA8y1x1LPC5aAVKRCfLf6o3YBkBjqhHk/sM3nhRS
P/TizPJhk9H9Z2vXUq6/aKtAQ6BXNVN48FP4YUIHZMbXb5tMOA1jrGKvNouicwoN
9SG9dKpN6nIDSdvHXx1iY8f93ZHsM+71bbRuMGjeyNYmsHVee7QHIJihdjK4TWxP
AgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFMmAd+BikoL1Rpzz
uvdMw964o605MA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAU18h
9bqwOlI5LJKwbADJ784g7wbylp7ppHR/ehb8t/W2+xUbP6umwHJdELFx7rxP462s
A20ucS6vxOOto70MEae0/0qyexAQH6dXQbLArvQsWdZHEIjzIVEpMMpghq9Gqx3t
OluwlN5E40EIosHsHdb9T7bWR9AUC8rmyrV7d35BH16Dx7aMOZawP5aBQW9gkOLo
+fsicdl9sz1Gv7SEr5AcD48Saq/v7h56rgJKihcrdv6sVIkkLE8/trKnToyokZf7
KcZ7XC25y2a2t6hbElGFtQl+Ynhw/qlqYLYdDnkM/crqJIByw5c/8nerQyIKx+u2
DISCLIBrQYoIwOula9+ZEsuK1V6ADJHgJgg2SMX6OBE1/yWDLfJ6v9r9jv6ly0Us
H8SIU653DtmadsWOLB2jutXsMq7Aqqz30XpN69QH4kj3Io6wpJ9qzo6ysmD0oyLQ
I+uUWnpp3Q+/QFesa1lQ2aOZ4W7+jQF5JyMV3pKdewlNWudLSDBaGOYKbeaP4NK7
5t98biGCwWg5TbSYWGZizEqQXsP6JwSxeRV0mcy+rSDeJmAc61ZRpqPq5KM/p/9h
3PFaTWwyI0PurKju7koSCTxdccK+efrCh2gdC/1cacwG0Jp9VJkqyTkaGa9LKkPz
Y11aWOIv4x3kqdbQCtCev9eBCfHJxyYNrJgWVqA=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFWTCCA0GgAwIBAgIBAjANBgkqhkiG9w0BAQsFADBOMQswCQYDVQQGEwJOTzEd
MBsGA1UECgwUQnV5cGFzcyBBUy05ODMxNjMzMjcxIDAeBgNVBAMMF0J1eXBhc3Mg
Q2xhc3MgMyBSb290IENBMB4XDTEwMTAyNjA4Mjg1OFoXDTQwMTAyNjA4Mjg1OFow
TjELMAkGA1UEBhMCTk8xHTAbBgNVBAoMFEJ1eXBhc3MgQVMtOTgzMTYzMzI3MSAw
HgYDVQQDDBdCdXlwYXNzIENsYXNzIDMgUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEB
BQADggIPADCCAgoCggIBAKXaCpUWUOOV8l6ddjEGMnqb8RB2uACatVI2zSRHsJ8Y
ZLya9vrVediQYkwiL944PdbgqOkcLNt4EemOaFEVcsfzM4fkoF0LXOBXByow9c3E
N3coTRiR5r/VUv1xLXA+58bEiuPwKAv0dpihi4dVsjoT/Lc+JzeOIuOoTyrvYLs9
tznDDgFHmV0ST9tD+leh7fmdvhFHJlsTmKtdFoqwNxxXnUX/iJY2v7vKB3tvh2PX
0DJq1l1sDPGzbjniazEuOQAnFN44wOwZZoYS6J1yFhNkUsepNxz9gjDthBgd9K5c
/3ATAOux9TN6S9ZV+AWNS2mw9bMoNlwUxFFzTWsL8TQH2xc519woe2v1n/MuwU8X
KhDzzMro6/1rqy6any2CbgTUUgGTLT2G/H783+9CHaZr77kgxve9oKeV/afmiSTY
zIw0bOIjL9kSGiG5VZFvC5F5GQytQIgLcOJ60g7YaEi7ghM5EFjp2CoHxhLbWNvS
O1UQRwUVZ2J+GGOmRj8JDlQyXr8NYnon74Do29lLBlo3WiXQCBJ31G8JUJc9yB3D
34xFMFbG02SrZvPAXpacw8Tvw3xrizp5f7NJzz3iiZ+gMEuFuZyUJHmPfWupRWgP
K9Dx2hzLabjKSWJtyNBjYt1gD1iqj6G8BaVmos8bdrKEZLFMOVLAMLrwjEsCsLa3
AgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFEe4zf/lb+74suwv
Tg75JbCOPGvDMA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAACAj
QTUEkMJAYmDv4jVM1z+s4jSQuKFvdvoWFqRINyzpkMLyPPgKn9iB5btb2iUspKdV
cSQy9sgL8rxq+JOssgfCX5/bzMiKqr5qb+FJEMwx14C7u8jYog5kV+qi9cKpMRXS
IGrs/CIBKM+GuIAeqcwRpTzyFrNHnfzSgCHEy9BHcEGhyoMZCCxt8l13nIoUE9Q2
HJLw5QY33KbmkJs4j1xrG0aGQ0JfPgEHU1RdZX33inOhmlRaHylDFCfChQ+1iHsa
O5S3HWCntZznKWlXWpuTekMwGwPXYshApqr8ZORK15FTAaggiG6cX0S5y2CBNOxv
033aSF/rtJC8LakcC6wc1aJoIIAE1vyxjy+7SjENSoYc6+I2KSb12tjE8nVhz36u
dmNKekBlk4f4HoCMhuWG1o8O/FMsYOgWYRqiPkN7zTlgVGr18okmAWiDSKIz6MkE
kbIRNBE+6tBDGR8Dk5AM/1E9V/RBbuHLoL7ryWPNbczk+DaqaJ3tvV2XcEQNtg41
3OEMXbugUZTLfhbrES+jkkXITHHZvMmZUldGL1DPvTVp9D0VzgalLA8+9oG6lLvD

u79leNKGef9JOxqDDPDeeOzI8k1MGt6CKfjBWtrt7uYnXuhF0J0cUahoq0Tj0Itq
4/g7u9xN12TyUb7mqqta6THuBrxzvxNiCp/HuZc=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFaTCCA1GgAwIBAgIJAJK4iNuwisFjMA0GCSqGSIb3DQEBCwUAMFIxCzAJBgNV
BAYTAlNLMRMwEQYDVQQHEwpCcmF0aXNsYXZhMRMwEQYDVQQKEwpEaXNpZyBhLnMu
MRkwFwYDVQQDExBDQSBEaXNpZyBSb290IFIyMB4XDTEyMDcxOTA5MTUzMFoXDTQy
MDcxOTA5MTUzMFowUjELMAkGA1UEBhMCU0sxEzARBgNVBAcTCkJyYXRpc2xhdmEx
EzARBgNVBAoTCkRpc2lnIGEucy4xGTAXBgNVBAMTEENBIERpc2lnIFJvb3QgUjIw
ggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCio8QACdaFXS1tFPbCw3Oe
NcJxVX6B+6tGUODBfEl45qt5WDza/3wcn9iXAng+a0EE6UG9vgMsRfYvZNSrXaNH
PWSb6WiaxswbP7q+sos0Ai6YVRn8jG+qX9pMzk0DIaPY0jSTVpbLTAwAFjxfGs3I
x2ymrdMxp7zo5eFm1tL7A7RBZckQrg4FY8aAamkw/dLukO8NJ9+flXP04SXabBbe
QTg06ov80egEFGEtQX6sx3dOy1FU+16SGBsEWmjGycT6txOgmLcRK7fWV8x8nhfR
yyX+hk4kLlYMeE2eARKmK6cBZW58Yh2EhN/qwGu1pSqVg8NTEQxzHQuyRpDRQjrO
QG6Vrf/GlK1ul4SOfW+eioANSW1z4nuSHsPzwfPrLgVv2RvPN3YEyLRa5Beny912
H9AZdugsBbPWnDTYltxhh5EF5EQIM8HauQhl1K6yNg3ruji6DOWbnuuNZt2Zz9aJ
QfYEkoopKW1rOhzndX0CcQ7zwOe9yxndnWCywmZgtrEE7snmhrmaZkCo5xHtgUUD
i/ZnWejBBhG93c+AAk9lQHhcR1DIm+YfgXvkRKhbhZri3lrVx/k6RGZL5DJUfORs
nLMOPReisjQS1n6yqEm70XooQL6iFh/f5DcfEXP7kAplQ6INfPgGAVUzfbANuPT1
rqVCV3w2EYx7XsQDnYx5nQIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1Ud
DwEB/wQEAwIBBjAdBgNVHQ4EFgQUtZn4r7CU9eMg1gqtzk5WpC5uQu0wDQYJKoZI
hvcNAQELBQADggIBACYGXnDnZTPIgm7ZnBc6G3pmsgH2eDtpXi/q/075KMOYKmFM
tCQSin1tERT3nLXK5ryeJ45MGcipvXrA1zYObYVybqjGom32+nNjf7xueQgcnYqf
GopTpti72TVVsRHFqQOzVju5hJMiXn7B9hJSi+osZ7z+Nkz1uM/Rs0mSO9MpDpkb
lvdhuDvEK7Z4bLQjb/D907JedR+Zlais9trhxTF7+9FGs9K8Z7RiVLoJ92Owk6Ka
+elSLotgEqv89WBW7xBci8QaQtyDW2QOy7W81k/BfDxujRNt+3vrMNDcTa/F1bal
TFtxyegxvug4BkihGuLq0t4SOVga/4AOgnXmt8kHbA7v/zjxmHHEt38OFdAlab0i
nSvtBfZGR6ztwPDUO+Ls7pZbkBNOHlY667DvlruWIxG68kOGdGSVyCh13x01utI3
gzhTODY7z2zp+WsO0PsE6E9312UBeIYMej4hYvF/Y3EMyZ9E26gnonW+boE+18Dr
G5gPcFw0sorMwIUY6256s/daoQe/qUKS82Ail+QUoQebTnbAjn39pCXHR+3/H3Os
zMOl6W8KjptlwlCFtaOgUxLMVYdh84GuEEZhvUQhuMI9dM9+JDX6HAcOmz0iyu8x
L4ysEr3vQCj8KWefshNPZiTEUxnpHikV7+ZtsH8tZ/3zbBt1RqPlShfppNcL
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFjTCCA3WgAwIBAgIEGErM1jANBgkqhkiG9w0BAQsFADBWMQswCQYDVQQGEwJD
TjEwMC4GA1UECgwnQ2hpbmEgRmluYW5jaWFsIENlcnRpZmljYXRpb24gQXV0aG9y
aXR5MRUwEwYDVQQDDAxDRkNBIEVWIFJPT1QwHhcNMTIwODA4MDMwNzAxWhcNMjkx
MjMxMDMwNzAxWjBWMQswCQYDVQQGEwJDTjEwMC4GA1UECgwnQ2hpbmEgRmluYW5j
aWFsIENlcnRpZmljYXRpb24gQXV0aG9yaXR5MRUwEwYDVQQDDAxDRkNBIEVWIFJP
T1QwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDXXWvNED8fBVnVBU03
sQ7smCuOFR36k0sXgiFxEFLXUWRwFsJVaU2OFW2fvwwbwuCjZ9YMrM8irq93VCpL
TIpTUnrD7i7es3ElweldPe6hL6P3KjzJIx1qqx2hp/Hz7KDVRM8Vz3IvHWOX6Jn5
/ZOkVIBMUtRSqy5J35DNuF++P96hyk0g1CXohClTt7GIH//62pCfCqktQT+x8Rgp

7hZZLDRJGqgG16iI0gNyejLi6mhNbiyWZXvKWfry4t3uMCz7zEasxGPrb382KzRz
EpR/38wmnvFyXVBlWY9ps4deMm/DGIq1lY+wejfeWkU7xzbh72fROdOXW3NiGUgt
hxwG+3SYIElz8AXSG7Ggo7cbcNOIabla1jj0Ytwli3i/+Oh+uFzJlU9fpy25IGvP
a931DfSCt/SyZi4QKPaXWnuWFo8BGS1sbn85WAZkgwGDg8NNkt0yxoekN+kWzqot
aK8KgWU6cMGbrU1tVMoqLUuFG7OA5nBFDWteNfB/O7ic5ARwiRIlk9oKmSJgamNg
TnYGmE69g60dWIolhdLHZR4tjsbftsbhf4oEIRUpdPA+nJCdDC7xij5aqgwJHsfV
PKPtl8MeNPo4+QgO48BdK4PRVmrJtqhUUy54Mmc9gn900PvhtgVguXDbjgv5E1hv
cWAQUhC5wUEJ73IfZzF4/5YFjQIDAQABo2MwYTAfBgNVHSMEGDAWgBTj/i39KNAL
tbq2osS/BqoFjJP7LzAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAd
BgNVHQ4EFgQU4/4t/SjQC7W6tqLEvwaqBYyT+y8wDQYJKoZIhvcNAQELBQADggIB
ACXGumvrh8vegjmWPfBEp2uEcwPenStPuiB/vHiyz5ewG5zz13ku9Ui20vsXiObT
ej/tUxPQ4i9qecsAIyjmHjdXNYmEwnZPNDatZ8POQQaIxffu2Bq41gt/UP+TqhdL
jOztUmCypAbqTuv0axn96/Ua4CUqmtzHQTb3yHQFhDmVOdYLO6Qn+gjYXB74BGBS
ESgoA//vU2YApUo0FmZ8/Qmkrp5nGm9BC2sGE5uPhnEFtC+NiWYzKXZUmhH4J/qy
P5Hgzg0b8zAarb8iXRvTvyUFTeGSGn+ZnzxEk8rUQElsgIfXBDrDMlI1Dlb4pd19
xIsNER9Tyx6yF7Zod1rg1MvIB671Oi6ON7fQAUtDKXeMOZePglr4UeWJoBjnaH9d
Ci77o0cOPaYjesYBx4/IXr9tgFa+iiS6M+qf4TIRnvHST4D2G0CvOJ4RUHlzEhLN
5mydLIhyPDCBBpEi6lmt2hkuIsKNuYyH4Ga8cyNfIWRjgEj1oDwYPZTISEEdQLpe
/v5WOaHIz16eGWRGENoXkbcFgKyLmZJ956LYBws2J+dIeWCKw9cTXPhyQN9Ky8+Z
AAoACxGV2lZFA4gKn2fQ1XmxqI1AbQ3CekD6819kR5LLU7m7Wc5P/dAVUwHY3+vZ
5nbv0CO7O6l5s9UCKc2Jo5YPSjXnTkLAdc0Hz+Ys63su
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIEHTCCAwWgAwIBAgIQToEtioJl4AsC7j41AkblPTANBgkqhkiG9w0BAQUFADCB
gTELMAkGA1UEBhMCR0IxGzAZBgNVBAgTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4G
A1UEBxMHU2FsZm9yZDEaMBgGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxJzAlBgNV
BAMTHkNPTU9ETyBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0wNjEyMDEwMDAw
MDBaFw0yOTEyMzEyMzU5NTlaMIGBMQswCQYDVQQGEwJHQjEbMBkGA1UECBMSR3Jl
YXRlciBNYW5jaGVzdGVyMRAwDgYDVQQHEwdTYWxmb3JkMRowGAYDVQQKExFDT01P
RE8gQ0EgTGltaXRlZDEnMCUGA1UEAxMeQ09NT0RPIENlcnRpZmljYXRpb24gQXV0
aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0ECLi3LjkRv3
UcEbVASY06m/weaKXTuH+7uIzg3jLz8GlvCiKVCZrts7oVewdFFxze1CkU1B/qnI
2GqGd0S7WWaXUF601CxwRM/aN5VCaTwwxHGzUvAhTaHYujl8HJ6jJJ3ygxaYqhZ8
Q5sVW7euNJH+1GImGEaaP+vB+fGQV+useg2L23IwambV4EajcNxo2f8ESIl33rXp
+2dtQem8Ob0y2WIC8bGoPW43nOIv4tOiJovGuFVDiOEjPqXSJDlqR6sA1KGzqSX+
DT+nHbrTUcELpNqsOO9VUCQFZUaTNE8tja3G1CEZ0o7KBWFxB3NH5YoZEr0ETc5O
nKVIrLsm9wIDAQABo4GOMIGLMB0GA1UdDgQWBBQLWOWLxkwVN6RAqTCpIb5HNlpW
/zAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zBJBgNVHR8EQjBAMD6g
PKA6hjhodHRwOi8vY3JsLmNvbW9kb2NhLmNvbS9DT01PRE9DZXJ0aWZpY2F0aW9u
QXV0aG9yaXR5LmNybDANBgkqhkiG9w0BAQUFAAOCAQEAPpiem/Yb6dc5t3iuHXIY
SdOH5EOC6z/JqvWote9VfCFSZfnVDeFs9D6Mk3ORLgLETgdxb8CPOGEIqB6BCsAv
IC9Bi5HcSEW88cbeunZrM8gALTFGTO3nnc+IlP8zwFboJIYmuNg4ON8qa90SzMc/
RxdMosIGlgnW2/4/PEZB31jiVg88O8EckzXZOFKs7sjsLjBOlDW0JB9LeGna8gI4
zJVSk/BwJVmcIGfE7vmLV2H0knZ9P4SNVbfo5azV8fUZVqZa+5Acr5Pr5RzUZ5dd
BA6+C4OmF4O5MBKgxTMVBbkN+8cFduPYSo38NBejxiEovjBFMR7HeL5YYTisO+IB
ZQ==

-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICiTCCAg+gAwIBAgIQH0evqmIAcFBUTAGem2OZKjAKBggqhkjOPQQDAzCBhTEL
MAkGA1UEBhMCR0IxGzAZBgNVBAgTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4GA1UE
BxMHU2FsZm9yZDEaMBgGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxKzApBgNVBAMT
IkNPTU9ETyBFQ0MgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMDgwMzA2MDAw
MDAwWhcNMzgwMTE4MjM1OTU5WjCBhTELMAkGA1UEBhMCR0IxGzAZBgNVBAgTEkdy
ZWF0ZXIgTWFuY2hlc3RlcjEQMA4GA1UEBxMHU2FsZm9yZDEaMBgGA1UEChMRQ09N
T0RPIENBIExpbWl0ZWQxKzApBgNVBAMTIkNPTU9ETyBFQ0MgQ2VydGlmaWNhdGlv
biBBdXRob3JpdHkwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAAQDR3svdcmCFYX7deSR
FtSrYpn1PlILBs5BAH+X4QokPB0BBO490o0JlwzgdeT6+3eKKvUDYEs2ixYjFq0J
cfRK9ChQtP6IHG4/bC8vCVlbpVsLM5niwz2J+Wos77LTBumjQjBAMB0GA1UdDgQW
BBR1cacZSBm8nZ3qQUfflMRId5nTeTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/
BAUwAwEB/zAKBggqhkjOPQQDAwNoADBlAjEA7wNbeqy3eApyt4jf/7VGFAkK+qDm
fQjGGoe9GKhzvSbKYAydzpmfz1wPMOG+FDHqAjAU9JM8SaczepBGR7NjfROb Trdv
GDeAU/7dIOA1mjbRxwG55tzd8/8dLDoWV9mSOdY=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF2DCCA8CgAwIBAgIQTKr5yttjb+Af907YWwOGnTANBgkqhkiG9w0BAQwFADCB
hTELMAkGA1UEBhMCR0IxGzAZBgNVBAgTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4G
A1UEBxMHU2FsZm9yZDEaMBgGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxKzApBgNV
BAMTIkNPTU9ETyBSU0EgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMTAwMTE5
MDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBhTELMAkGA1UEBhMCR0IxGzAZBgNVBAgT
EkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4GA1UEBxMHU2FsZm9yZDEaMBgGA1UEChMR
Q09NT0RPIENBIExpbWl0ZWQxKzApBgNVBAMTIkNPTU9ETyBSU0EgQ2VydGlmaWNh
dGlvbiBBdXRob3JpdHkwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCR
6FSS0gpWsawNJN3Fz0RndJkrN6N9I3AAcbxT38T6KhKPS38QVr2fcHK3YX/JSw8X
pz3jsARh7v8Rl8f0hj4K+j5c+ZPmNHrZFGvnnLOFoIJ6dq9xkNfs/Q36nGz637CC
9BR++b7Epi9Pf5l/tfxnQ3K9DADWietrLNPtj5gcFKt+5eNu/Nio5JIk2kNrYrhV
/erBvGy2i/MOjZrkm2xpmfh4SDBF1a3hDTxFYPwyllEnvGfDyi62a+pGx8cgoLEf
Zd5ICLqkTqnyg0Y3hOvozIFIQ2dOciqbXL1MGyiKXCJ7tKuY2e7gUYPDCUZObT6Z
+pUX2nwzV0E8jVHtC7ZcryxjGt9XyD+86V3Em69FmeKjWiS0uqlWPc9vqv9JWL7w
qP/0uK3pN/u6uPQLOvnoQ0IeidiEyxPx2bvhiWC4jChWrBQdnArncevPDt09qZah
SL0896+1DSJMwBGB7FY79tOi4lu3sgQiUpWAk2nojkxl8ZEDLXB0AuqLZxUpaVIC
u9ffUGpVRr+goyhhf3DQw6KqLCGqR84onAZFdr+CGCe01a60y1Dma/RMhnEw6abf
Fobg2P9A3fvQQoh/ozM6LlweQRGBY84YcWsr7KaKtzFcOmpH4MN5WdYgGq/yapiq
crxXStJLnbsQ/LBMQeXtHT1eKJ2czL+zUdqnR+WEUwIDAQABo0IwQDAdBgNVHQ4E
FgQUu69+Aj36pvE8hI6t7jiY7NkyMtQwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB
/wQFMAMBAf8wDQYJKoZIhvcNAQEMBQADggIBAArx1UaEt65Ru2yyTUEUAJNMnMvl
wFTPoCWOAvn9sKIN9SCYPBMtrFaisNZ+EZLpLrqeLppysb0ZRGxhNaKatBYSaVqM
4dc+pBroLwP0rmEdEBsqpIt6xf4FpuHA1sj+nq6PK7o9mfjYcwlYRm6mnPTXJ9OV
2jeDchzTc+CiR5kDOF3VSXkAKRzH7JsgHAckaVd4sjn8OoSgtZx8jb8uk2Intzna
FxiuvTwJaP+EmzzV1gsD41eeFPfR60/IvYcjt7ZJQ3mFXLrrkguhxuhoqEwWsRqZ
CuhTLJK7oQkYdQxlqHvLI7cawiiFwxv/0Cti76R7CZGYZ4wUAc1oBmpjIXUDgIiK

boHGhfKppC3n9KUkEEeDys30jXlYsQab5xoq2Z0B15R97QNKyvDb6KkBPvVWmcke
jkk9u+UJueBPSZI9FoJAzMxZxuY67RIuaTxslbH9qh17f4a+Hg4yRvv7E491f0yL
S0Zj/gA0QHDBw7mh3aZw4gSzQbzpgJHqZJx64SIDqZxubw5lT2yHh17zbqD5daWb
QOhTsiedSrnAdyGN/4fy3ryM7xfft0kL0fJuMAsaDk527RH89elWsn2/x20Kk4yl
0MC2Hb46TpSi125sC8KKfPog88Tk5c0NqMuRkrF8hey1FGlmDoLnzc7ILaZRfyHB
NVOFBkpdn627G190
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIB9zCCAX2gAwIBAgIQBiUzsUcDMydc+Y2aub/M+DAKBggqhkjOPQQDAzA9MQsw
CQYDVQQGEwJVUzESMBAGA1UEChMJQ2VydGFpbmx5MRowGAYDVQQDExFDZXJ0YWlu
bHkgUm9vdCBFMTAeFw0yMTA0MDEwMDAwMDBaFw00NjA0MDEwMDAwMDBaMD0xCzAJ
BgNVBAYTAlVTMRIwEAYDVQQKEwlDZXJ0YWlubHkxGjAYBgNVBAMTEUNlcnRhaW5s
eSBSb290IEUxMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAE3m/4fxzf7flHh4axpMCK
+IKXgOqPyEpeKn2IaKcBYhSRJHpcnqMXfYqGITQYUBsQ3tA3SybHGWCA6TS9YBk2
QNYphwk8kXr2vBMj3VlOBF7PyAIcGFPBMdjaIOlEjeR2o0IwQDAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU8ygYy2R17ikq6+2uI1g4
hevIIgcwCgYIKoZIzj0EAwMDaAAwZQIxALGOWiDDshliTd6wT99u0nCK8Z9+aozm
ut6Dacpps6kFtZaSF4fC0urQe87YQVt8rgIwRt7qy12a7DLCZRawTDBcMPPaTnOG
BtjOiQRINzf43TNRnXCve1XYAS59BWQOhriR
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFRzCCAy+gAwIBAgIRAI4P+UuQcWhlM1T01EQ5t+AwDQYJKoZIhvcNAQELBQAw
PTELMAkGA1UEBhMCVVMxEjAQBgNVBAoTCUNlcnRhaW5seTEaMBgGA1UEAxMRQ2Vy
dGFpbmx5IFJvb3QgUjEwHhcNMjEwNDAxMDAwMDAwWhcNNDYwNDAxMDAwMDAwWjA9
MQswCQYDVQQGEwJVUzESMBAGA1UEChMJQ2VydGFpbmx5MRowGAYDVQQDExFDZXJ0
YWlubHkgUm9vdCBSMTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBANA2
1B/q3avk0bbm+yLA3RMNansiExyXPGhjZjKcA7WNpIGD2ngwEc/csiu+kr+O5MQT
vqRoTNoCaBZ0vrLdBORrKt03H2As2/X3oXyVtwxwhi7xOu9S98zTm/mLvg7fMbed
aFySpvXl8wo0tf97ouSHocavFwDvA5HtqRxOcT3Si2yJ9HiG5mpJoM610rCrm/b0
1C7jcvk2xusVtyWMOvwlDbMicyF0yEqWYZL1LwsYpfSt4u5BvQF5+paMjRcCMLT5
r3gajLQ2EBAHBXDQ9DGQilHFhiZ5shGIXsXwClTNSaa/ApzSRKft43jvRl5tcdF5
cBxGX1HpyTfcX35pe0HfNEXgO4T0oYoKNp43zGJS4YkNKPl6I7ENPT2a/Z2B7yyQ
wHtETrtJ4A5KVpK8y7XdeReJkd5hiXSSqOMyhb5OhaRLWcsrxXiOcVTQAjeZjOVJ
6uBUcqQRBi8LjMFbvrWhsFNunLhgkR9Za/kt9JQKl7XsxXYDVBtlUrpMklZRNaBA
2CnbrlJ2Oy0wQJuK0EJWtLeIAaSHO1OWzaMWj/Nmqhexx2DgwUMFDO6bW2BvBlyH
Wyf5QBGenDPBt+U1VwV/J84XIIwc/PH72jEpSe31C4SnT8H2TsIonPru4K8H+zMR
eiFPCyEQtkA6qyI6BJyLm4SGcprSp6XEtHWRqSsjAgMBAAGjQjBAMA4GA1UdDwEB
/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBTgqj8ljZ9EXME66C6u
d0yEPmcM9DANBgkqhkiG9w0BAQsFAAOCAgEAuVevuBLaV4OPaAszHQNTVfSVcOQr
PbA56/qJYv331hgELyE03fFo8NWWWt7CgKPBjcZq91l3rhVkz1t5BXdm6ozTaw3d
8VkswTOlMIAVRQdFGjEitpIAq5lNOo93r6kiyi9jyhXWx8bwPWz8HA2YEGGeEaIi
1wrykXprOQ4vMMM2SZ/g6Q8CRFA3lFV96p/2O7qUpUzpvD5RtOjKkjZUbVwlKNrd
rRT90+7iIgXr0PK3aBLXWopBGsaSpVo7Y0VPv+E6dyIvXL9G+VoDhRNCX8reU9di
taY1BMJH/5n9hN9czulegChB8n3nHpDYT3Y+gjwN/KUD+nsa2UUeYNrEjvn8K8l7

lcUq/6qJ34IxD3L/DCfXCh5WAFAeDJDBlrXYFIW7pw0WwfgHJBu6haEaBQmAupVj
yTrsJZ9/nbqkRxWbRHDxakvWOF5D8xh+UG7pWijmZeZ3Gzr9Hb4DJqPb1OG7fpYn
Kx3upPvaJVQTA945xsMfTZDsjxtK0hzthZU4UHlG1sGQUDGpXJpuHfUzVounmdLy
yCwzk5Iwx06MZTMQZBf9JBeW0Y3COmor6xOLRPIh80oat3df1+2IpHLlOR+Vnb5n
wXARPbv0+Em34yaXOp/SX3z7wJl8OSngex2/DaeP0ik0biQVy96QXr8axGbqwua6
OV+KmalBWQewLK8=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDqDCCApCgAwIBAgIJAP7c4wEPyUj/MA0GCSqGSIb3DQEBBQUAMDQxCzAJBgNV
BAYTAkZSMRIwEAYDVQQKDAlEaGlteW90aXMxETAPBgNVBAMMCENlcnRpZ25hMB4X
DTA3MDYyOTE1MTMwNVoXDTI3MDYyOTE1MTMwNVowNDELMAkGA1UEBhMCRlIxEjAQ
BgNVBAoMCURoaW15b3RpczERMA8GA1UEAwwIQ2VydGlnbmEwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDIaPHJ1tazNHUmgh7stL7qXOEm7RFHYeGifBZ4
QCHkYJ5ayGPhxLGWkv8YbWkj4Sti993iNi+RB7lIzw7sebYs5zRLcAglozyHGxny
gQcPOJAZ0xH+hrTy0V4eHpbNgGzOOzGTtvKg0KmVEn2lmsxryIRWijOp5yIVUxbw
zBfsV1/pogqYCd7jX5xv3EjjhQsVWqa6n6xI4wmy9/Qy3l40vhx4XUJbzg4ij02Q
130yGLMLLGq/jj8UEYkgDncUtT2UCIf3JR7VsmAA7G8qKCVuKj4YYxclPz5EIBb2
JsglrgVKtOdjLPOMFlN+XPsRGgjBRmKfIrjxwo1p3Po6WAbfAgMBAAGjgbwwgbkw
DwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUGu3+QTmQtCRZvgHyUtVF9lo53BEw
ZAYDVR0jBF0wW4AUGu3+QTmQtCRZvgHyUtVF9lo53BGhOKQ2MDQxCzAJBgNVBAYT
AkZSMRIwEAYDVQQKDAlEaGlteW90aXMxETAPBgNVBAMMCENlcnRpZ25hggkA/tzj
AQ/JSP8wDgYDVR0PAQH/BAQDAgEGMBEGCWCGSAGG+EIBAQQEAwIABzANBgkqhkiG
9w0BAQUFAAOCAQEAhQMeknH2Qq/ho2Ge6/PAD/Kl1NqV5ta+aDY9fm4fTIrv0Q8h
bV6lUmPOEvjvKtpv6zf+EwLHyzs+ImvaYS5/1HI93TDhHkxAGYwP15zRgzB7mFnc
fca5DClMoTOi62c6ZYTTluLtdkVwj7Ur3vkj1kluPBS1xp81HlDQwY9qcEQCYsuu
HWhBp6pX6FOqB9IG9tUUBguRA3UsbHK1YZWaDYu5Def131TN3ubY1gkIl2PlwS6w
t0QmwCbAr1UwnjvVNioZBPRcHv/PLLf/0P2HQBHVESO7SMAhqaQoLf0V+LBOK/Qw
WyH8EZE0vkHve52Xdf+XlcCWWC/qu0bXu+TZLg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIGWzCCBEOgAwIBAgIRAMrpG4nxVQMNo+ZBbcTjpuEwDQYJKoZIhvcNAQELBQAw
WjELMAkGA1UEBhMCRlIxEjAQBgNVBAoMCURoaW15b3RpczEcMBoGA1UECwwTMDAw
MiA0ODE0NjMwODEwMDAzNjEZMBcGA1UEAwwQQ2VydGlnbmEgUm9vdCBDQTAeFw0x
MzEwMDEwODMyMjdaFw0zMzEwMDEwODMyMjdaMFoxCzAJBgNVBAYTAkZSMRIwEAYD
VQQKDAlEaGlteW90aXMxHDAaBgNVBAsMEzAwMDIgNDgxNDYzMDgxMDAwMzYxGTAX
BgNVBAMMEENlcnRpZ25hIFJvb3QgQ0EwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAw
ggIKAoICAQDNGDllGlmx6mQWDoyUJJV8g9PFOSbcDO8WV43X2KyjQn+Cyu3NW9sO
ty3tRQgXstmzy9YXUnIo245Onoq2C/mehJpNdt4iKVzSs9IGPjA5qXSjklYcoW9M
CiBtnyN6tMbaLOQdLNyzKNAT8kxOAkmhVECe5uUFoC2EyP+YbNDrihqECB63aCPu
I9Vwzm1RaRDuoXrC0SIxwoKF0vJVdlB8JXrJhFwLrN1CTivngqIkicuQstDuI7pm
TLtipPlTWmR7fJj6o0ieD5Wupxj0auwuA0Wv8HT4Ks16XdG+RCYyKfHx9WzMfgIh
C59vpD++nVPiz32pLHxYGpfhPTc3GGYo0kDFUYqMwy3OU4gkWGQwFsWq4NYKpkDf
ePb1BHxpE4S80dGnBs8B92jAqFe7OmGtBIyT46388NtEbVncSVmurJqZNjBBe3Yz
IoejwpKGbvlw7q6Hh5UbxHq9MfPU0uWZ/75I7HX1eBYdpnDBfzwboZL7z8g81sWT

Co/1VTp2lc5ZmIoJlXcymoO6LAQ6l73UL77XbJuiyn1tJslV1c/DeVIICZkHJC1k
JWumIWmbat10TWuXekG9qxf5kBdIjzb5LdXF2+6qhUVB+s06RbFo5jZMm5BX7CO5
hwjCxAnxl4YqKE3idMDaxIzb3+KhF1nOJFl0Mdp//TBt2dzhauH8XwIDAQABo4IB
GjCCARYwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYE
FBiHVuBud+4kNTxOc5of1uHieX4rMB8GA1UdIwQYMBaAFBiHVuBud+4kNTxOc5of
1uHieX4rMEQGA1UdIAQ9MDswOQYEVR0gADAxMC8GCCsGAQUFBwIBFiNodHRwczov
L3d3d3cuY2VydGlnbmEuZnIvYXV0b3JpdGVzLzBtBgNVHR8EZjBkMC+gLaArhilo
dHRwOi8vY3JsLmNlcnRpZ25hLmZyL2NlcnRpZ25hcm9vdGNhLmNybDAxoC+gLYYr
aHR0cDovL2NybC5kaGlteW90aXMuY29tL2NlcnRpZ25hcm9vdGNhLmNybDANBgkq
hkiG9w0BAQsFAAOCAgEAlLieT/DjlQgi581oQfccVdV8AOItOoldaDgvUSILSo3L
6btdPrtcPbEo/uRTVRPPoZAbAh1fZkYJMyjhDSSXcNMQH+pkV5a7XdrnxIxPTGRG
HVyH41neQtGbqH6mid2PHMkwgu07nM3A6RngatgCdTer9zQoKJHyBApPNeNgJgH6
0BGM+RFq7q89w1DTj18zeTyGqHNFkIwgtnJzFyO+B2XleJINugHA64wcZr+shncB
lA2c5uk5jR+mUYyZDDl34bSb+hxnV29qao6pK0xXeXpXIs/NX2NGjVxZOob4Mkdi
o2cNGJHc+6Zr9UhhcyNZjgKnvETq9Emd8VRY+WCv2hikLyhF3HqgiIZd8zvn/yk1
gPxkQ5Tm4xxvvq0OKmOZK8l+hfZx6AYDlf7ej0gcWtSS6Cvu5zHbugRqh5jnxV/v
faci9wHYTfmJ0A6aBVmknpjZbyvKcL5kwlWj9Omvw5Ip3IgWJJk8jSaYtlu3zM63
Nwf9JtmYhST/WSMDmu2dnajkXjjO11INb9I/bbEFa0nOipFGc/T2L/Coc3cOZayh
jWZSaX5LaAzHHjcng6WMxwLkFM1JAbBzs/3GkDpv0mztO+7skb6iQ12LAEpmJURw
3kAP+HwV96LOPNdeE4yBFxgX0b3xdxA61GU5wSesVywlVP+i2k+KYTlerj1KjL0=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICZTCCAeugAwIBAgIQeI8nXIESUiClBNAt3bpz9DAKBggqhkjOPQQDAzB0MQsw
CQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNvIERhdGEgU3lzdGVtcyBTLkEuMScw
JQYDVQQLEx5DZXJ0dW0gQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxGTAXBgNVBAMT
EENlcnR1bSBFQy0zODQgQ0EwHhcNMTgwMzI2MDcyNDU0WhcNNDMwMzI2MDcyNDU0
WjB0MQswCQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNvIERhdGEgU3lzdGVtcyBT
LkEuMScwJQYDVQQLEx5DZXJ0dW0gQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxGTAX
BgNVBAMTEENlcnR1bSBFQy0zODQgQ0EwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAATE
KI6rGFtqvm5kN2PkzeyrOvfMobgOgknXhimfoZTy42B4mIF4Bk3y7JoOV2CDn7Tm
Fy8as10CW4kjPMIRBSqniBMY81CE1700LCeJVf/OTOffph8oxPBUw7l8t1Ot68Kj
QjBAMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFI0GZnQkdjrzife81r1HfS+8
EF9LMA4GA1UdDwEB/wQEAwIBBjAKBggqhkjOPQQDAwNoADBlAjADVS2m5hjEfO/J
UG7BJw+ch69u1RsIGL2SKcHvlJF40jocVYli5RsJHrpka/F2tNQCMQC0QoSZ/6vn
nvuRlydd3LBbMHHOXjgaatkl5+r3YZJW+OraNsKHZZYuciUvf9/DE8k=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDuzCCAqOgAwIBAgIDBETAMA0GCSqGSIb3DQEBBQUAMH4xCzAJBgNVBAYTAlBM
MSIwIAYDVQQKExlVbml6ZXRvIFRlY2hub2xvZ2llcyBTLkEuMScwJQYDVQQLEx5D
ZXJ0dW0gQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxIjAgBgNVBAMTGUNlcnR1bSBU
cnVzdGVkIE5ldHdvcmsgQ0EwHhcNMDgxMDIyMTIwNzM3WhcNMjkxMjMxMTIwNzM3
WjB+MQswCQYDVQQGEwJQTDEiMCAGA1UEChMZVW5pemV0byBUZWNobm9sb2dpZXMg
Uy5BLjEnMCUGA1UECxMeQ2VydHVtIENlcnRpZmljYXRpb24gQXV0aG9yaXR5MSIw
IAYDVQQDExlDZXJ0dW0gVHJ1c3RlZCBOZXR3b3JrIENBMIIBIjANBgkqhkiG9w0B

AQEFAAOCAQ8AMIIBCgKCAQEA4/t9o3K6wvDJFIf1awFO4W5AB7ptJ11/91sts1rH
UV+rpDKmYYe2bg+G0jACl/jXaVehGDldamR5xgFZrDwxSjh80gTSSyjoIF87B6LM
TXPb865Px1bVWqeWifrzq2jUI4ZZJ88JJ7ysbnKDHDBy3+Ci6dLhdHUZvSqeexVU
BBvXQzmtVSjF4hq79MDkrjhJM8x2hZ85RdKknvISjFH4fOQtf/WsX+sWn7Et0brM
kUJ3TCXJkDhv2/DM+44el1k+1WBO5gUo7Ul5E0u6SNsv+XLTOcr+H9g0cvW0QM8x
AcPs3hEtF10fuFDRXhmnad4HMyjKUJX5p1TLVIZQRan5SQIDAQABo0IwQDAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBBQIds3LB/8k9sXN7buQvOKEN0Z19zAOBgNV
HQ8BAf8EBAMCAQYwDQYJKoZIhvcNAQEFBQADggEBAKaorSLOAT2mo/9i0Eidi15y
sHhE49wcrwn9I0j6vSrEuVUEtRCjjSfeC4Jj0O7eDDd5QVsisrCaQVymcODU0HfL
I9MA4GxWL+FpDQ3Zqr8hgVDZBqWo/5U30Kr+4rP1mS1FhIrlQgnXdAIv94nYmem8
J9RHjboNRhx3zxSkHLmkMcScKHQDNP8zGSal6Q10tz6XxnboJ5ajZt3hrvJBW8qY
VoNzcOSGGtIxQbovvi0TWnZvTuhOgQ4/WwMioBK+ZlgRSssDxLQqKi2WF+A5VLxI
03YnnZotBqbJ7DnSq9ufmgsnAjUpsUCV5/nonFWIGUbWtzT1fs45mtk48VH3Tyw=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF0jCCA7qgAwIBAgIQIdbQSk8lD8kyN/yqXhKN6TANBgkqhkiG9w0BAQ0FADCB
gDELMAkGA1UEBhMCUEwxIjAgBgNVBAoTGVVuaXpldG8gVGVjaG5vbG9naWVzIFMu
QS4xJzAlBgNVBAsTHkNlcnR1bSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTEkMCIG
A1UEAxMbQ2VydHVtIFRydXN0ZWQgTmV0d29yayBDQSAyMCIYDzIwMTExMDA2MDgz
OTU2WhgPMjA0NjEwMDYwODM5NTZaMIGAMQswCQYDVQQGEwJQTDEiMCAGA1UEChMZ
VW5pemV0byBUZWNobm9sb2dpZXMgUy5BLjEnMCUGA1UECxMeQ2VydHVtIENlcnRp
ZmljYXRpb24gQXV0aG9yaXR5MSQwIgYDVQQDExtDZXJ0dW0gVHJ1c3RlZCBOZXR3
b3JrIENBIDIwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC9+Xj45tWA
DGSdhhuWZGc/IjoedQF97tcZ4zJzFxrqZHmuULlIEub2pt7uZld2ZuAS9eEQCsn
0+i6MLs+CRqnSZXvK0AkwpfHp+6bJe+oCgCXhVqqndwpyeI1B+twTUrWwbNWuKFB
OJvR+zF/j+Bf4bE/D44WSWDXBo0Y+aomEKsq09DRZ40bRr5HMNUuctHFY9rnY3lE
fktjJImGLjQ/KUxSiyqnwOKRKIm5wFv5HdnnJ63/mgKXwcZQkpsCLL2puTRZCr+E
Sv/f/rOf69me4Jgj7KZrdxYq28ytOxykh9xGc14ZYmhFV+SQgkK7QtbwYeDBoz1m
o130GO6IyY0XRSmZMnUCMe4pJshrAua1YkV/NxVaI2iJ1D7eTiew8EAMvE0Xy02i
sx7QBlrd9pPPV3WZ9fqGGmd4s7+W/jTcvedSVuWz5XV710GRBdxdaeOVDUO5/IOW
OZV7bIBaTxNyxtd9KXpEulKkKtVBRgkg/iKgtlswjbyJDNXXcPiHUv3a76xRLgez
Tv7QCdpw75j6VuZt27VXS9zlLCUVyJ4ueE742pyehizKV/Ma5ciSixqClnrDvFAS
adgOWkaLOusm+iPJtrCBvkIApPjW/jAux9JG9uWOdf3yzLnQh1vMBhBgu4M1t15n
3kfsmUjxpKEV/q2MYo45VU85FrmxY53/twIDAQABo0IwQDAPBgNVHRMBAf8EBTAD
AQH/MB0GA1UdDgQWBBS2oVQ5AsOgP46KvPrU+Bym0ToO/TAOBgNVHQ8BAf8EBAMC
AQYwDQYJKoZIhvcNAQENBQADggIBAHGlDs7k6b8/ONWJWsQCYftMxRQXLYtPU2sQ
F/xlhMcQSZDe28cmk4gmb3DWAl45oPePq5a1pRNcgRRtDoGCERuKTsZPpd1iHkTf
CVn0W3cLN+mLIMb4Ck4uWBzrM9DPhmDJ2vuAL55MYIR4PSFk1vtBHxgP58l1cb29
XN40hz5BsA72udY/CROWFC/emh1auVbONTqwX3BNXuMp8SMoclm2q8KMZiYcdywm
djWLKKdpoPk79SPdhRB0yZADVpHnr7pH1BKXESLjokmUbOe3lEu6LaTaM4tMpkT/
WjzGHWTYtTHkpjx6qFcL2+1hGsvxznN3Y6SHb0xRONbkX8eftoEq5IVIeVheO/jb
AoJnwTnbw3RLPTYe+SmTiGhbqEQZIfCn6IENLOiTNrQ3ssqwGyZ6miUfmpqAnksq
P/ujmv5zMnHCnsZy4YpoJ/HkD7TETKVhk/iXEAcqMCWpuchxuO9ozC1+9eB+D4Ko
b7a6bINDd82Kkhehnlt4Fj1F4jNy3eFmypnTycUm/Q1oBEauttmbjL4ZvrHG8hnj
XALKLNhvSgfZyTXaQHXyxKcZb55CEJh15pWLYLztxRLXis7VmFxWlgPF7ncGNf/P
5O4/E2Hu29othfDNrp2yGAlFw5Khchf8R7agCyzxxN5DaAhqXzvwdmP7zAYspsbi

DrW5viSP
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFwDCCA6igAwIBAgIQHr9ZULjJgDdMBvfrVU+17TANBgkqhkiG9w0BAQ0FADB6
MQswCQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNvIERhdGEgU3lzdGVtcyBTLkEu
MScwJQYDVQQLEx5DZXJ0dW0gQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxHzAdBgNV
BAMTFkNlcnR1bSBUcnVzdGVkIFJvb3QgQ0EwHhcNMTgwMzE2MTIxMDEzWhcNNDMw
MzE2MTIxMDEzWjB6MQswCQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNvIERhdGEg
U3lzdGVtcyBTLkEuMScwJQYDVQQLEx5DZXJ0dW0gQ2VydGlmaWNhdGlvbiBBdXRo
b3JpdHkxHzAdBgNVBAMTFkNlcnR1bSBUcnVzdGVkIFJvb3QgQ0EwggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQDRLY67tzbqbTeRn06TpwXkKQMlzhyC93yZ
n0EGze2jusDbCSzBfN8pfktlL5On1AFrAygYo9idBcEq2EXxkd7fO9CAAozPOA/q
p1x4EaTByIVcJdPTsuclzxFUl6s1wB52HO8AU5853BSlLCIls3Jy/I2z5T4IHhQq
NwuIPMqw9MjCoa68wb4pZ1Xi/K1ZXP69VyywkI3C7Te2fJmItdUDmj0VDT06qKhF
8JVOJVkdzZhpu9PMMsmN74H+rX2Ju7pgE8pllWeg8xn2A1bUatMn4qGtg/BKEiJ3
HAVz4hlxQsDsdUaakFjgao4rpUYwBI4Zshfjvqm6f1bxJAPXsiEodg42MEx51UGa
mqi4NboMOvJEGyCI98Ul1z3G4z5D3Yf+xOr1Uz5MZf87Sst4WmsXXw3Hw09Omiqi
7VdNIuJGmj8PkTQkfVXjjJU30xrwCSss0smNtA0Aq2cpKNgB9RkEth2+dv5yXMSF
ytKAQd8FqKPVhJBPC/PgP5sZ0jeJP/J7UhyM9uH3PAeXjA6iWYEMspA90+NZRu0P
qafegGtaqge2Gcu8V/OXIXoMsSt0Puvap2ctTMSYnjYJdmZm/Bo/6khUHL4wvYBQ
v3y1zgD2DGHZ5yQD4OMBgQ692IU0iL2yNqh7XAjlRICMb/gv1SHKHRzQ+8S1h9E6
Tsd2tTVItQIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBSM+xx1
vALTn04uSNn5YFSqxLNP+jAOBgNVHQ8BAf8EBAMCAQYwDQYJKoZIhvcNAQENBQAD
ggIBAEii1QALLtA/vBzVtVRJHlpr9OTy4EA34MwUe7nJ+jW1dReTagVphZzNTxl4
WxmB82M+w85bj/UvXgF2Ez8sALnNllI5SW0ETsXpD4YN4fqzX4IS8TrOZgYkNCvo
zMrnadyHncI013nR03e4qllY/p0m+jiGPp2Kh2RX5Rc64vmNueMzeMGQ2Ljdt4NR
5MTMI9UGfOZR0800McD2RrsLrfw9EAUqO0qRJe6M1ISHgCq8CYyqOhNf6DR5UMEQ
GfnTKB7U0VEwKbOukGfWHwpjscWpxkIxYxeU72nLL/qMFH3EQxiJ2fAyQOaA4kZf
5ePBAFmo+eggvIksDkc0C+pXwlM2/KfUrzHN/gLldfq5Jwn58/U7yn2fqSLLiMmq
0Uc9NneoWWRrJ8/vJ8HjJLWG965+Mk2weWjROeiQWMODvA8s1pfrzgzhIMfatz7D
P78v3DSk+yshzWePS/Tj6tQ/50+6uaWTRRxmHyH6ZF5v4HaUMst19W7l9o/HuKTM
qJZ9ZPskWkoDbGs4xugDQ5r3V7mzKWmTOPQD8rv7gmsHINFSH5pkAnuYZttcTVoP
0ISVoDwUQwbKytu4QTbaakRnh6+v40URFWkIsr4WOZckbxJF0WddCajJFdr60qZf
E2Efv4WstK2tBZQIgx51F9NxO5NQI1mg7TyRVJ12AMXDuDjb
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICHTCCAaOgAwIBAgIUQ3CCd89NXTTxyq4yLzf39H91oJ4wCgYIKoZIzj0EAwMw
TjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTErMCkGA1UEAwwiQ29t
bVNjb3BlIFB1YmxpYyBUcnVzdCBFQ0MgUm9vdC0wMTAeFw0yMTA0MjgxNzM1NDNa
Fw00NjA0MjgxNzM1NDJaME4xCzAJBgNVBAYTAlVTMRIwEAYDVQQKDAlDb21tU2Nv
cGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgRUNDIFJvb3QtMDEw
djAQBgcqhkjOPQIBBgUrgQQAIgNiAARLNumuV16ocNfQj3Rid8NeeqrltqLxeP0C
flfdkXmcbLlSiFS8LwS+uM32ENEp7LXQoMPwiXAZu1FlxUOcw5tjnSCDPgYLpkJE
hRGnSjot6dZoL0hOUysHP029uax3OVejQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYD

VR0PAQH/BAQDAgEGMB0GA1UdDgQWBBSOB2LAUN3GGQYARnQE9/OufXVNMDAKBggq
hkjOPQQDAwNoADBlAjEAnDPfQeMjqEI2Jpc1XHvr20v4qotzVRVcrHgpD7oh2MSg
2NED3W3ROT3Ek2DS43KyAjB8xX6I01D1HiXo+k515liWpDVfG2XqYZpwI7UNo5uS
Um9poIyNStDuiw7LR47QjRE=
-----END CERTIFICATE-----

-----BEGIN
CERTIFICATE-----
MIICHDCCAaOgAwIBAgIUKP2ZYEFHpgE6yhR7H+/5aAiDXX0wCgYIKoZIzj0EAwMw
TjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTErMCkGA1UEAwwiQ29t
bVNjb3BlIFB1YmxpYyBUcnVzdCBFQ0MgUm9vdC0wMjAeFw0yMTA0MjgxNzQ0NTRa
Fw00NjA0MjgxNzQ0NTNaME4xCzAJBgNVBAYTAlVTMRIwEAYDVQQKDAlDb21tU2Nv
cGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgRUNDIFJvb3QtMDIw
djAQBgcqhkjOPQIBBgUrgQQAIgNiAAR4MIHoYx7l63FRD/cHB8o5mXxO1Q/MMDAL
j2aTPs+9xYa9+bG3tD60B8jzljHz7aRP+KNOjSkVWLjVb3/ubCK1sK9IRQq9qEmU
v4RDsNuESgMjGWdqb8FuvAY5N9GIIvejQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgEGMB0GA1UdDgQWBBTmGHX/72DehKT1RsfeSlXjMjZ59TAKBggq
hkjOPQQDAwNnADBkAjAmc0l6tqvmSfR9Uj/UQQSugEODZXW5hYA4O9Zv5JOGq4/n
ich/m35rChJVYaoR4HkCMHfoMXGsPHED1oQmHhS48zs73u1Z/GtMMH9ZzkXpc2AV
mkzw5l4lIhVtwodZ0LKOag==
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFbDCCA1SgAwIBAgIUPgNJgXUWdDGOTKvVxZAplsU5EN0wDQYJKoZIhvcNAQEL
BQAwTjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTErMCkGA1UEAwwi
Q29tbVNjb3BlIFB1YmxpYyBUcnVzdCBSU0EgUm9vdC0wMTAeFw0yMTA0MjgxNjQ1
NTRaFw00NjA0MjgxNjQ1NTNaME4xCzAJBgNVBAYTAlVTMRIwEAYDVQQKDAlDb21t
U2NvcGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgUlNBIFJvb3Qt
MDEwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCwSGWjDR1C45FtnYSk
YZYSwu3D2iM0GXb26v1VWvZVAVMP8syMl0+5UMuzAURWlv2bKOx7dAvnQmtVzslh
suitQDy6uUEKBU8bJoWPQ7VAtYXR1HHcg0Hz9kXHgKKEUJdGzqAMxGBWBB0HW0al
DrJLpA6lfO741GIDuZNqihS4cPgugkY4Iw50x2tBt9Apo52AsH53k2NC+zSDO3Oj
WiE260f6GBfZumbCk6SP/F2krfxQapWsvCQz0b2If4b19bJzKo98rwjyGpg/qYFl
P8GMicWWMJoKz/TUyDTtnS+8jTiGU+6Xn6myY5QXjQ/cZip8UlF1y5mO6D1cv547
KI2DAg+pn3LiLCuz3GaXAEDQpFSOm117RTYm1nJD68/A6g3czhLmfTifBSeolz7p
UcZsBSjBAg/pGG3svZwG1KdJ9FQFa2ww8esD1eo9anbCyxooSU1/ZOD6K9pzg4H/
kQO9lLvkuI6cMmPNn7togbGEW682v3fuHX/3SZtS7NJ3Wn2RnU3COS3kuoL4b/JO
Hg9O5j9ZpSPcPYeoKFgo0fEbNttPxP/hjFtyjMcmAyejOQoBqsCyMWCDIqFPEgkB
Ea801M/XrmLTBQe0MXXgDW1XT2mH+VepuhX2yFJtocucH+X8eKg1mp9BFM6ltM6U
CBwJrVbl2rZJmkrqYxhTnCwuwwIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4G
A1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUN12mmnQywsL5x6YVEFm45P3luG0wDQYJ
KoZIhvcNAQELBQADggIBAK+nz97/4L1CjU3lIpbfaOp9TSp90K09FlxD533Ahuh6
NWPxzIHIxgvoLlI1pKZJkGNRrDSsBTtXAOnTYtPZKdVUvhwQkZyybf5Z/Xn36lbQ
nmhUQo8mUuJM3y+Xpi/SB5io82BdS5pYV4jvguX6r2yBS5KPQJqTRlnLX3gWsWc+
QgvfKNmwrZggvkN80V4aCRckjXtdlemrwWCrWxhkgPut4AZ9HcpZuPN4KWfGVh2v
trV0KnahP/t1MJ+UXjulYPPLXAziDslg+MkfFoom3ecnf+slpoq9uC02EJqxWE2a
aE9gVOX2RhOOiKy8IUISrcZKiX2bwdgt6ZYD9KJ0DLwAHb/WNyVntHKLr4W96ioD
j8z7PEQkguIBpQtZtjSNMgsSDesnwv1B10A8ckYpwIzqug/xBpMu95yo9GA+o/E4

Xo4TwbM6l4c/ksp4qRyv0LAbJh6+cOx69TOY6lz/KwsETkPdY34Op054A5U+1C0w
lREQKC6/oAI+/15Z0wUOlV9TRe9rh9VIzRamloPh37MG88EU26fsHItdkJANclHn
YfkUyq+Dj7+vsQpZXdxc1+SWrVtgHdqul7I52Qb1dgAT+GhMIbA1xNxVssnBQVoc
icCMb3SgazNNtQEo/a2tiRc7ppqEvOuM6sRxJKi6KfkIsidWNTJf6jn7MZrVGczw
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFbDCCA1SgAwIBAgIUVBa/O345lXGN0aoApYYNK496BU4wDQYJKoZIhvcNAQEL
BQAwTjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTErMCkGA1UEAwwi
Q29tbVNjb3BlIFB1YmxpYyBUcnVzdCBSU0EgUm9vdC0wMjAeFw0yMTA0MjgxNzE2
NDNaFw00NjA0MjgxNzE2NDJaME4xCzAJBgNVBAYTAlVTMRIwEAYDVQQKDAlDb21t
U2NvcGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgUlNBIFJvb3Qt
MDIwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDh+g77aAASyE3VrCLE
NQE7xVTlWXZjpX/rwcRqmL0yjReA61260WI9JSMZNRTpf4mnG2I81lDnNJUDMrG0
kyI9p+Kx7eZ7Ti6Hmw0zdQreqjXnfuU2mKKuJZ6VszKWpCtYHu8//mI0SFHRtI1C
rWDaSWqVcN3SAOLMV2MCe5bdSZdbkk6V0/nLKR8YSvgBKtJjCW4k6YnS5cciTNxz
hkcAqg2Ijq6FfUrpuzNPDlJwnZXjfG2WWy09X6GDRl224yW4fKcZgBzqZUPckXk2
LHR88mcGyYnJ27/aaL8j7dxrrSiDeS/sOKUNNwFnJ5rpM9kzXzehxfCrPfp4sOcs
n/Y+n2Dg70jpkEUeBVF4GiwSLFworA2iI540jwXmojPOEXcT1A6kHkIfhs1w/tku
FT0du7jyU1fbzMZ0KZwYszZ1OC4PVKH4kh+Jlk+71O6d6Ts2QrUKOyrUZHk2EOH5
kQMreyBUzQ0ZGshBMjTRsJnhkB4BQDa1t/qp5Xd1pCKBXbCL5CcSD1SIxtuFdOa3
wNemKfrb3vOTlycEVS8KbzfFPROvCgCpLIscgSjX74Yxqa7ybrjKaixUR9gqiC6v
wQcQeKwRoi9C8DfF8rhW3Q5iLc4tVn5V8qdE9isy9COoR+jUKgF4z2rDN6ieZdIs
5fq6M8EGRPbmz6UNp2YINIos8wIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4G
A1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUR9DnsSL/nSz12Vdgs7GxcJXvYXowDQYJ
KoZIhvcNAQELBQADggIBAIZpsU0v6Z9PIpNojuQhmaPORVMbc0RTAIFhzTHjCLqB
KCh6krm2qMhDnscTJk3C2OVVnJJdUNjCK9v+5qiXz1I6JMNlZFxHMaNlNRPDk7n3
+VGXu6TwYofF1gbTl4MgqX67tiHCpQ2EAOHyJxCDut0DgdXdaMNmEMjRdrSzbyme
APnCKfWxkxlSaRosTKCL4BWaMS/TiJVZbuXEs1DIFAhKm4sTg7GkcrI7djNB3Nyq
pgdvHSQSn8h2vS/ZjvQs7rfSOBAkNlEv41xdgSGn2rtO/+YHqP65DSdsu3BaVXoT
6fEqSWnHX4dXTEN5bTpl6TBcQe7rd6VzEojov32u5cSoHw2OHG1QAk8mGEPej1WF
sQs3BWDJVTkSBKEqz3EWnzZRSb9wO55nnPt7eck5HHisd5FUmrh1CoFSl+NmYWvt
PjgelmFV4ZFUjO2MJB+ByRCac5krFk5yAD9UG/iNuovnFNa2RU9g7Jauwy8CTl2d
lklyALKrdVwPaFsdZcJfMw8eD/A7hvWwTruc9+olBdytoptLFwG+Qt81IR2tq670
v64fG9PiO/yzcnMcmyiQiRM9HcEARwmWmjgb3bHPDcK0RPOWlc4yOo80nOAXx17O
rg3bhzjlP1v9mxnhMUF6cKojawHhRUzNlM47ni3niAIi9G7oyOzWPPO5std3eqx7
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIC2zCCAmCgAwIBAgIQfMmPK4TX3+oPyWWa00tNljAKBggqhkjOPQQDAzBIMQsw
CQYDVQQGEwJERTEVMBMGA1UEChMMRC1UcnVzdCBHbWJIMSIwIAYDVQQDExlELVRS
VVNUIEJSIFJvb3QgQ0EgMSAyMDIwMB4XDTIwMDIxMTA5NDUwMFoXDTM1MDIxMTA5
NDQ1OVowSDELMAkGA1UEBhMCREUxFTATBgNVBAoTDEQtVHJ1c3QgR21iSDEiMCAG
A1UEAxMZRC1UUlVTVCBCUiBSb290IENBIDEgMjAyMDB2MBAGByqGSM49AgEGBSuB
BAAiA2IABMbLxyjR+4T1mu9CFCDhQ2tuda38KwOE1HaTJddZO0Flax7mNCq7dPYS
zuht56vkPE4/RAiLzRZxy7+SmfSk1zxQVFKQhYN4lGdnoxwJGT11NIXe7WB9xwy0

QVK5buXuQqOCAQ0wggEJMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFHOREKv/
VbNafAkl1bK6CKBrqx9tMA4GA1UdDwEB/wQEAwIBBjCBxgYDVR0fBIG+MIG7MD6g
PKA6hjhodHRwOi8vY3JsLmQtdHJ1c3QubmV0L2NybC9kLXRydXN0X2JyX3Jvb3Rf
Y2FfMV8yMDIwLmNybDB5oHegdYZzbGRhcDovL2RpcmVjdG9yeS5kLXRydXN0Lm5l
dC9DTj1ELVRSVVNUJTIwQlIlMjBSb290JTIwQ0ElMjAxJTIwMjAyMCxPPUQtVHJ1
c3QlMjBHbWJILEM9REU/Y2VydGlmaWNhdGVyZXZvY2F0aW9ubGlzdDAKBggqhkjO
PQQDAwNpADBmAjEAlJAtE/rhY/hhY+ithXhUkZy4kzg+GkHaQBZTQgjKL47xPoFW
wKrY7RjEsK70PvomAjEA8yjixtsrmfu3Ubgko6SUeho/5jbiA1czijDLgsfWFBHV
dWNbFJWcHwHP2NVypw87
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFqTCCA5GgAwIBAgIQczswBEhb2U14LnNLyaHcZjANBgkqhkiG9w0BAQ0FADBI
MQswCQYDVQQGEwJERTEVMBMGA1UEChMMRC1UcnVzdCBHbWJIMSIwIAYDVQQDExlE
LVRSVVNUIEJSIFJvb3QgQ0EgMiAyMDIzMB4XDTIzMDUwOTA4NTYzMVoXDTM4MDUw
OTA4NTYzMFowSDELMAkGA1UEBhMCREUxFTATBgNVBAoTDEQtVHJ1c3QgR21iSDEi
MCAGA1UEAxMZRC1UUlVTVCBCUiBSb290IENBIDIgMjAyMzCCAiIwDQYJKoZIhvcN
AQEBBQADggIPADCCAgoCggIBAK7/CVmRgApKaOYkP7in5Mg6CjoWzckjYaCTcfKr
i3OPoGdlYNJUa2NRb0kz4HIHE304zQaSBylSa053bATTlfrdTIzZXcFhfUvnKLNE
gXtRr90zsWh81k5M/itoucpmacTsXld/9w3HnDY25QdgrMBM6ghs7wZ8T1soegj8
k12b9py0i4a6Ibn08OhZWiihNIQaJZG2tY/vsvmA+vk9PBFy2OMvhnbFeSzBqZCT
Rphny4NqoFAjpzv2gTng7fC5v2Xx2Mt6++9zA84A9H3X4F07ZrjcjrqDy4d2A/wl
2ecjbwb9Z/Pg/4S8R7+1FhhGaRTMBffb00msa8yr5LULQyReS2tNZ9/WtT5PeB+U
cSTq3nD88ZP+npNa5JRal1QMNXtfbO4AHyTsA7oC9Xb0n9Sa7YUsOCIvx9gvdhFP
/Wxc6PWOJ4d/GUohR5AdeY0cW/jPSoXk7bNbjb7EZChdQcRurDhaTyN0dKkSw/bS
uREVMweR2Ds3OmMwBtHFIjYoYiMQ4EbMl6zWK11kJNXuHA7e+whadSr2Y23OC0K+
0bpwHJwh5Q8xaRfX/Aq03u2AnMuStIv13lmiWAmlY0cL4UEyNEHZmrHZqLAbWt4N
DfTisl01gLmB1IRpkQLLddCNxbU9CZEJjxShFHR5PtbJFR2kWVki3PaKRT08EtY+
XTIvAgMBAAGjgY4wgYswDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUZ5Dw1t61
GNVGKX5cq/ieCLxklRAwDgYDVR0PAQH/BAQDAgEGMEkGA1UdHwRCMEAwPqA8oDqG
OGh0dHA6Ly9jcmwuZC10cnVzdC5uZXQvY3JsL2QtdHJ1c3RfYnJfcm9vdF9jYV8y
XzIwMjMuY3JsMA0GCSqGSIb3DQEBDQUAA4ICAQA097N3U9swFrktpSHxQCF16+tI
FoE9c+CeJyrrd6kTpGoKWloUMz1oH4Guaf2Mn2VsNELZLdB/eBaxOqwjMa1ef67n
riv6uvw8l5VAk1/DLQOj7aRvU9f6QA4w9QAgLABMjDu0ox+2v5Eyq6+SmNMW5tTR
VFxDWy6u71cqqLRvpO8NVhTaIasgdp4D/Ca4nj8+AybmTNudX0KEPUUDAxxZiMrc
LmEkWqTqJwtzEr5SswrPMhfiHocaFpVIbVrg0M8JkiZmkdijYQ6qgYF/6FKC0ULn
4B0Y+qSFNueG4A3rvNTJ1jxD8V1Jbn6Bm2m1iWKPiFLY1/4nwSPFyysCu7Ff/vtD
hQNGvl3GyiEm/9cCnnRK3PgTFbGBVzbLZVzRHTF36SXDw7IyN9XxmAnkbWOACKsG
koHU6XCPpz+y7YaMgmo1yEJagtFSGkUPFaUA8JR7ZSdXOUPPfH/mvTWze/EZTN46
ls/pdu4D58JDUjxqgejBWoC9EV2Ta/vH5mQ/u2kc6d0li690yVRAysuTEwrt+2aS
Ecr1wPrYg1UDfNPFIkZ1cGt5SAYqgpq/5usWDiJFAbzdNpQ0qTUmiteXue4Icr80
knCDgKs4qllo3UCkGJCy89UDyibK79XH4I9TjvAA46jtn/mtd+ArY0+ew+43u3gJ
hJ65bvspmZDogNOfJA==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIC2zCCAmCgAwIBAgIQXwJB13qHfEwDo6yWjfv/0DAKBggqhkjOPQQDAzBIMQsw
CQYDVQQGEwJERTEVMBMGA1UEChMMRC1UcnVzdCBHbWJIMSIwIAYDVQQDExlELVRS
VVNUIEVWIFJvb3QgQ0EgMSAyMDIwMB4XDTIwMDIxMTEwMDAwMFoXDTM1MDIxMTA5
NTk1OVowSDELMAkGA1UEBhMCREUxFTATBgNVBAoTDEQtVHJ1c3QgR21iSDEiMCAG
A1UEAxMZRC1UUlVTVCBFViBSb290IENBIDEgMjAyMDB2MBAGByqGSM49AgEGBSuB
BAAiA2IABPEL3YZDIBnfl4XoIkqbz52Yv7QFJsnL46bSj8WeeHsxiamJrSc8ZRCC
/N/DnU7wMyPE0jL1HLDfMxddxfCxivnvubcUyilKwg+pf3VlSSowZ/Rk99Yad9rD
wpdhQntJraOCAQ0wggEJMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFH8QARY3
OqQo5FD4pPfsazK2/umLMA4GA1UdDwEB/wQEAwIBBjCBxgYDVR0fBIG+MIG7MD6g
PKA6hjhodHRwOi8vY3JsLmQtdHJ1c3QubmV0L2NybC9kLXRydXN0X2V2X3Jvb3Rf
Y2FfMV8yMDIwLmNybDB5oHegdYZzbGRhcDovL2RpcmVjdG9yeS5kLXRydXN0Lm5l
dC9DTj1ELVRSVVNUJTIwRVYlMjBSb290JTIwQ0ElMjAxJTIwMjAyMCxPPUQtVHJ1
c3QlMjBHbWJILEM9REU/Y2VydGlmaWNhdGVyZXZvY2F0aW9ubGlzdDAKBggqhkjO
PQQDAwNpADBmAjEAyjzGKnXCXnViOTYAYFqLwZOZzNnbQTs7h5kXO9XMT8oi96CA
y/m0sRtW9XLS/BnRAjEAkfcwkz8QRitxpNA7RJvAKQIFskF3UfN5Wp6OFKBOQtJb
gfM0agPnIjhQW+0ZT0MW
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFqTCCA5GgAwIBAgIQaSYJfoBLTKCnjHhiU19abzANBgkqhkiG9w0BAQ0FADBI
MQswCQYDVQQGEwJERTEVMBMGA1UEChMMRC1UcnVzdCBHbWJIMSIwIAYDVQQDExlE
LVRSVVNUIEVWIFJvb3QgQ0EgMiAyMDIzMB4XDTIzMDUwOTA5MTAzM1oXDTM4MDUw
OTA5MTAzMlowSDELMAkGA1UEBhMCREUxFTATBgNVBAoTDEQtVHJ1c3QgR21iSDEi
MCAGA1UEAxMZRC1UUlVTVCBFViBSb290IENBIDIgMjAyMzCCAiIwDQYJKoZIhvcN
AQEBBQADggIPADCCAgoCggIBANiOo4mAC7JXUtypU0w3uX9jFxPvp1sjW2l1sJkK
F8GLxNuo4MwxusLyzV3pt/gdr2rElYfXR8mV2IIEUD2BCP/kPbOx1sWy/YgJ25yE
7CUXFId/MHibaljJtnMoPDT3mfd/06b4HEV8rSyMlD/YZxBTfiLNTiVR8CUkNRFe
EMbsh2aJgWi6zCudR3Mfvc2RpHJqnKIbGKBv7FD0fUDCqDDPvXPIEysQEx6Lmqg6
lHPTGGkKSv/BAQP/eX+1SH977ugpbzZMlWGG2Pmic4ruri+W7mjNPU0oQvlFKzIb
RlUWaqZLKfm7lVa/Rh3sHZMdwGWyH6FDrlaeoLGPaxK3YG14C8qKXO0elg6DpkiV
jTujIcSuWMYAsoS0I6SWhjW42J7YrDRJmGOVxcttSEfi8i4YHtAxq9107PncjLgc
jmgjutDzUNzPZY9zOjLHfP7KgiJPvo5iR2blzYfi6NUPGJ/lBHJLRjwQ8kTCZFZx
TnXonMkmdMV9WdEKWw9t/p51HBjGGjp82A0EzM23RWV6sY+4roRIPrN6TagD4uJ+
ARZZaBhDM7DS3LAaQzXupdqpRlyuhoFBAUp0JuyfBr/CBTdkdXgpaP3F9ev+R/nk
hbDhezGdpn9yo7nELC7MmVcOIQxFAZRl62UJxmMiCzNJkkg8/M3OsD6Onov4/knF
NXJHAgMBAAGjgY4wgYswDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUqvyREBuH
kV8Wub9PS5FeAByxMoAwDgYDVR0PAQH/BAQDAgEGMEkGA1UdHwRCMEAwPqA8oDqG
OGh0dHA6Ly9jcmwuZC10cnVzdC5uZXQvY3JsL2QtdHJ1c3RfZXZfcm9vdF9jYV8y
XzIwMjMuY3JsMA0GCSqGSIb3DQEBDQUAA4ICAQCTy6UfmRHsmg1fLBWTxj++EI14
QvBukEdHjqOSMo1wj/Zbjb6JzkcBahsgIIlbyIIQbODnmaprxiqgYzWRaoUlrRc4
pZt+UPJ26oUFKidBK7GB0aL2QHWpDsvxVUjY7NHss+jOFKE17MJeNRqrphYBBo7q
3C+jisosketSjl8MmxfPy3MHGcRqwnNU73xDUmPBEcrCRbH0O1P1aa4846XerOhU
t7KR/aypH/KH5BfGSah82ApB9PI+53c0BFLd6IHyTS9URZ0V4U/M5d40VxDJI3IX
cI1QcB9WbMy5/zpaT2N6w25lBx2Eof+pDGOJbbJAiDnXH3dotfyc1dZnaVuodNv8
ifYbMvekJKZ2t0dT741Jj6m2g1qllpBFYfXeA08mD6iL8AOWsKwV0HFaanuU5nCT
2vFp4LJiTZ6P/4mdm13NRemUAiKN4DV/6PEEeXFsVIP4M7kFMhtYVRFP0OUnR3Hs
7dpn1mKmS00PaaLJvOwiS5THaJQXfuKOKD62xur1NGyfN4gHONuGcfrNlUhDbqNP

gofXNJhuS5N5YHVpD/Aa1VP6IQzCP+k/HxiMkl14p3ZnGbuy6n/pcAlWVqOwDAst
Nl7F6cTVg8uGF5csbBNvh1qvSaYd2804BC5f4ko1Di1L+KIkBI3Y4WNeApI02phh
XBxvWHZks/wCuPWdCg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIEMzCCAxugAwIBAgIDCYPzMA0GCSqGSIb3DQEBCwUAME0xCzAJBgNVBAYTAkRF
MRUwEwYDVQQKDAxELVRydXN0IEdtYkgxJzAlBgNVBAMMHkQtVFJVU1QgUm9vdCBD
bGFzcyAzIENBIDIgMjAwOTAeFw0wOTExMDUwODM1NThaFw0yOTExMDUwODM1NTha
ME0xCzAJBgNVBAYTAkRFMRUwEwYDVQQKDAxELVRydXN0IEdtYkgxJzAlBgNVBAMM
HkQtVFJVU1QgUm9vdCBDbGFzcyAzIENBIDIgMjAwOTCCASIwDQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBANOySs96R+91myP6Oi/WUEWJNTrGa9v+2wBoqOADER03
UAifTUpolDWzU9GUY6cgVq/eUXjsKj3zSEhQPgrfRlWLJ23DEE0NkVJD2IfgXU42
tSHKXzlABF9bfsyjxiupQB7ZNoTWSPOSHjRGICTBpFGOShrvUD9pXRl/RcPHAY9R
ySPocq60vFYJfxLLHLGvKZAKyVXMD9O0Gu1HNVpK7ZxzBCHQqr0ME7UAyiZsxGsM
lFqVlNpQmvH/pStmMaTJOKDfHR+4CS7zp+hnUquVH+BGPtikw8paxTGA6Eian5Rp
/hnd2HN8gcqW3o7tszIFZYQ05ub9VxC1X3a/L7AQDcUCAwEAAaOCARowggEWMA8G
A1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFP3aFMSfMN4hvR5COfyrYyNJ4PGEMA4G
A1UdDwEB/wQEAwIBBjCB0wYDVR0fBIHLMIHIMIGAoH6gfIZ6bGRhcDovL2RpcmVj
dG9yeS5kLXRydXN0Lm5ldC9DTj1ELVRSVVNUJTIwUm9vdCUyMENsYXNzJTIwMyUy
MENBJTIwMiUyMDIwMDksTz1ELVRydXN0JTIwR21iSCxDPURFP2NlcnRpZmljYXRl
cmV2b2NhdGlvbmxpc3QwQ6BBoD+GPWh0dHA6Ly93d3cuZC10cnVzdC5uZXQvY3Js
L2QtdHJ1c3Rfcm9vdF9jbGFzc18zX2NhXzJfMjAwOS5jcmwwDQYJKoZIhvcNAQEL
BQADggEBAH+X2zDI36ScfSF6gHDOFBJpiBSVYEQBrLLpME+bUMJm2H6NMLVwMeni
acfzcNsgFYbQDfC+rAF1hM5+n02/t2A7nPPKHeJeaNijnZflQGDSNiH+0LS4F9p0
o3/U37CYAqxva2ssJSRyoWXuJVrl5jLn8t+rSfrzkGkj2wTZ51xY/GXUl77M/C4K
zCUqNQT4YJEVdT1B/yMfGchs64JTBKbkTCJNjYy6zltz7GRUUG3RnFX7acM2w4y8
PIWmawomDeCTmGCufsYkl4phX5GOZpIJhzbNi5stPvZR1FDUWSi9g/LMKHtThm3Y
Johw1+qRzT65ysCQblrGXnRl11z+o+I=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIEQzCCAyugAwIBAgIDCYP0MA0GCSqGSIb3DQEBCwUAMFAxCzAJBgNVBAYTAkRF
MRUwEwYDVQQKDAxELVRydXN0IEdtYkgxKjAoBgNVBAMMIUQtVFJVU1QgUm9vdCBD
bGFzcyAzIENBIDIgRVYgMjAwOTAeFw0wOTExMDUwODUwNDZaFw0yOTExMDUwODUw
NDZaMFAxCzAJBgNVBAYTAkRFMRUwEwYDVQQKDAxELVRydXN0IEdtYkgxKjAoBgNV
BAMMIUQtVFJVU1QgUm9vdCBDbGFzcyAzIENBIDIgRVYgMjAwOTCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAJnxhDRwui+3MKCOvXwEz75ivJn9gpfSegpn
ljgJ9hBOlSJzmY3aFS3nBfwZcyK3jpgAvDw9rKFs+9Z5JUut8Mxk2og+KbgPCdM0
3TP1YtHhzRnp7hhPTFiu4h7WDFsVWtg6uMQYZB7jM7K1iXdODL/ZlGsTl28So/6Z
qQTMFexgaDbtCHu39b+T7WYxg4zGcTSHThfqr4uRjRxWQa4iN1438h3Z0S0NL2lR
p75mpoo6Kr3HGrHhFPC+Oh25z1uxav60sUYgovseO3Dvk5h9jHOW8sXvhXCtKSb8
HgQ+HKDYD8tSg2J87otTlZCpV6LqYQXY+U3EJ/pure3511H3a6UCAwEAAaOCASQw
ggEgMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFNOUikxiEyoZLsyvcop9Ntea
HNxnMA4GA1UdDwEB/wQEAwIBBjCB3QYDVR0fBIHVMIHSMIGHoIGEoIGBhn9sZGFw
Oi8vZGlyZWN0b3J5LmQtdHJ1c3QubmV0L0NOPUQtVFJVU1QlMjBSb290JTIwQ2xh

c3MlMjAzJTIwQ0ElMjAyJTIwRVYlMjAyMDA5LE89RC1UcnVzdCUyMEdtYkgsQz1E
RT9jZXJ0aWZpY2F0ZXJldm9jYXRpb25saXN0MEagRKBChkBodHRwOi8vd3d3LmQt
dHJ1c3QubmV0L2NybC9kLXRydXN0X3Jvb3RfY2xhc3NfM19jYV8yX2V2XzIwMDku
Y3JsMA0GCSqGSIb3DQEBCwUAA4IBAQA07XtaPKSUiO8aEXUHL7P+PPoeUSbrh/Yp
3uDx1MYkCenBz1UbtDDZzhr+BlGmFaQt77JLvyAoJUnRpjZ3NOhk31KxEcdzes05
nsKtjHEh8lprr988TlWvsoRlFIm5d8sqMb7Po23Pb0iUMkZv53GMoKaEGTcH8gNF
CSuGdXzfX2lXANtu2KZyIktQ1HWYVt+3GP9DQ1CuekR78HlR10M9p9OB0/DJT7na
xpeG0ILD5EJt/rDiZE4OJudANCa1CInXCGNjOCd1HjPqbqjdn5lPdE2BiYBL3ZqX
KVwvvoFBuYz/6n1gBp7N1z3TLqMVvKjmJuVvw9y4AyHqnxbxLFS1
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDtzCCAp+gAwIBAgIQDOfg5RfYRv6P5WD8G/AwOTANBgkqhkiG9w0BAQUFADBl
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDExtEaWdpQ2VydCBBc3N1cmVkIElEIFJv
b3QgQ0EwHhcNMDYxMTEwMDAwMDAwWhcNMzExMTEwMDAwMDAwWjBlMQswCQYDVQQG
EwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNl
cnQuY29tMSQwIgYDVQQDExtEaWdpQ2VydCBBc3N1cmVkIElEIFJvb3QgQ0EwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCtDhXO5EOAXLGH87dg+XESpa7c
JpSIqvTO9SA5KFhgDPiA2qkVlTJhPLWxKISKityfCgyDF3qPkKyK53lTXDGEKvYP
mDI2dsze3Tyoou9q+yHyUmHfnyDXH+Kx2f4YZNISW1/5WBg1vEfNoTb5a3/UsDg+
wRvDjDPZ2C8Y/igPs6eD1sNuRMBhNZYW/lmci3Zt1/GiSw0r/wty2p5g0I6QNcZ4
VYcgoc/lbQrISXwxmDNsIumH0DJaoroTghHtORedmTpyoeb6pNnVFzF1roV9Iq4/
AUaG9ih5yLHa5FcXxH4cDrC0kqZWs72yl+2qp/C3xag/lRbQ/6GW6whfGHdPAgMB
AAGjYzBhMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBRF66Kv9JLLgjEtUYunpyGd823IDzAfBgNVHSMEGDAWgBRF66Kv9JLLgjEtUYun
pyGd823IDzANBgkqhkiG9w0BAQUFAAOCAQEAog683+Lt8ONyc3pklL/3cmbYMuRC
dWKuh+vy1dneVrOfzM4UKLkNl2BcEkxY5NM9g0lFWJc1aRqoR+pWxnmrEthngYTf
fwk8lOa4JiwgvT2zKIn3X/8i4peEH+ll74fg38FnSbNd67IJKusm7Xi+fT8r87cm
NW1fiQG2SVufAQWbqz0lwcy2f8Lxb4bG+mRo64EtlOtCt/qMHt1i8b5QZ7dsvfPx
H2sMNgcWfzd8qVttevESRmCD1ycEvkvOl77DZypoEd+A5wwzZr8TDRRu838fYxAe
+o0bJW1sj6W3YQGx0qMmoRBxna3iw/nDmVG3KwcIzi7mULKn+gpFL6Lw8g==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDljCCAn6gAwIBAgIQC5McOtY5Z+pnI7/Dr5r0SzANBgkqhkiG9w0BAQsFADBl
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDExtEaWdpQ2VydCBBc3N1cmVkIElEIFJv
b3QgRzIwHhcNMTMwODAxMTIwMDAwWhcNMzgwMTE1MTIwMDAwWjBlMQswCQYDVQQG
EwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNl
cnQuY29tMSQwIgYDVQQDExtEaWdpQ2VydCBBc3N1cmVkIElEIFJvb3QgRzIwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDZ5ygvUj82ckmIkzTz+GoeMVSA
n61UQbVH35ao1K+ALbkKz3X9iaV9JPrjIgwrvJUXCzO/GU1BBpAAvQxNEP4Htecc
biJVMWWXvdMX0h5i89vqbFCMP4QMls+3ywPgym2hFEwbid3tALBSfK+RbLE4E9Hp
EgjAALAcKxHad3A2m67OeYfcgnDmCXRwVWmvo2ifv922ebPynXApVfSr/5Vh88lA
bx3RvpO704gqu52/clpWcTs/1PPRCv4o76Pu2ZmvA9OPYLfykqGxvYmJHzDNw6Yu

YjOuFgJ3RFrngQo8p0Quebg/BLxcoIfhG69Rjs3sLPr4/m3wOnyqi+RnlTGNAgMB
AAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQW
BBTOw0q5mVXyuNtgv6l+vVa1lzan1jANBgkqhkiG9w0BAQsFAAOCAQEAyqVVjOPI
QW5pJ6d1Ee88hjZv0p3GeDgdaZaikmkuOGybfQTUiaWxMTeKySHMq2zNixya1r9I
0jJmwYrA8y8678Dj1JGG0VDjA9tzd29KOVPt3ibHtX2vK0LRdWLjSisCx1BL4Gni
lmwORGYQRI+tBev4eaymG+g3NJ1TyWGqolKvSnAWhsI6yLETcDbYz+70CjTVW0z9
B5yiutkBclzzTcHdDrEcDcRjvq30FPuJ7KJBDkzMyFdA0G4Dqs0MjomZmWzwPDCv
ON9vvKO+KSAnq3T/EyJ43pdSVR6DtVQgA+6uwE9W3jfMw3+qBCe703e4YtsXfJwo
IhNzbM8m9Yop5w==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICRjCCAc2gAwIBAgIQC6Fa+h3foLVJRK/NJKBs7DAKBggqhkjOPQQDAzBlMQsw
CQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cu
ZGlnaWNlcnQuY29tMSQwIgYDVQQDExtEaWdpQ2VydCBBc3N1cmVkIElEIFJvb3Qg
RzMwHhcNMTMwODAxMTIwMDAwWhcNMzgwMTE1MTIwMDAwWjBlMQswCQYDVQQGEwJV
UzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQu
Y29tMSQwIgYDVQQDExtEaWdpQ2VydCBBc3N1cmVkIElEIFJvb3QgRzMwdjAQBgcq
hkjOPQIBBgUrgQQAIgNiAAQZ57ysRGXtzbg/WPuNsVepRC0FFfLvC/8QdJ+1YlJf
Zn4f5dwbRXkLzMZTCp2NXQLZqVneAlr2lSoOjThKiknGvMYDOAdfVdp+CW7if17Q
RSAPWXYQ1qAk8C3eNvJsKTmjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/
BAQDAgGGMB0GA1UdDgQWBBTL0L2p4ZgFUaFNN6KDec6NHSrkhDAKBggqhkjOPQQD
AwNnADBkAjAlpIFFAmsSS3V0T8gj43DydXLefInwz5FyYZ5eEJJZVrmDxxDnOOlY
JjZ91eQ0hjkCMHw2U/Aw5WJjOpnitqM7mzT6HtoQknFekROn3aRukswy1vUhZscv
6pZjamVFkpUBtA==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAeFw0wNjExMTAwMDAwMDBaFw0zMTExMTAwMDAwMDBaMGExCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IENBMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUKKPC3eQyaKl7hLOllsB
CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97
nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt
43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P
T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4
gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR
TLtm8KPiGxvDl7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDl7I90VUw
DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr
hMAtudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg
06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5BjfIttep3Sp+dWOIrWcBAI+0tKIJF
PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls

YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4dzp30d8tbQk
CAUw7C29C79Fv1C5qfPrmAESrciIxpg0X40KPMbp1ZWVbd4=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDjjCCAnagAwIBAgIQAzrx5qcRqaC7KGSxHQn65TANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBH
MjAeFw0xMzA4MDExMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IEcyMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuzfNNNx7a8myaJCtSnX/RrohCgiN9RlUyfuI
2/Ou8jqJkTx65qsGGmvPrC3oXgkkRLpimn7Wo6h+4FR1IAWsULecYxpsMNzaHxmx
1x7e/dfgy5SDN67sH0NO3Xss0r0upS/kqbitOtSZpLYl6ZtrAGCSYP9PIUkY92eQ
q2EGnI/yuum06ZIya7XzV+hdG82MHauVBJVJ8zUtluNJbd134/tJS7SsVQepj5Wz
tCO7TG1F8PapspUwtP1MVYwnSlcUfIKdzXOS0xZKBgyMUNGPHgm+F6HmIcr9g+UQ
vIOlCsRnKPZzFBQ9RnbDhxSJITRNrw9FDKZJobq7nMWxM4MphQIDAQABo0IwQDAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUTiJUIBiV
5uNu5g/6+rkS7QYXjzkwDQYJKoZIhvcNAQELBQADggEBAGBnKJRvDkhj6zHd6mcY
1Yl9PMWLSn/pvtsrF9+wX3N3KjITOYFnQoQj8kVnNeyIv/iPsGEMNKSuIEyExtv4
NeF22d+mQrvHRAiGfzZ0JFrabA0UWTW98kndth/Jsw1HKj2ZL7tcu7XUIOGZX1NG
Fdtom/DzMNU+MeKNhJ7jitralj41E6Vf8PlwUHBHQRFXGU7Aj64GxJUTFy8bJZ91
8rGOmaFvE7FBcf6IKshPECBV1/MUReXgRPTqh5Uykw7+U0b6LJ3/iyK5S9kJRaTe
pLiaWN0bfVKfjllDiIGknibVb63dDcY3fe0Dkhvld1927jyNxF1WW6LZZm6zNTfl
MrY=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICPzCCAcWgAwIBAgIQBVVWvPJepDU1w6QP1atFcjAKBggqhkjOPQQDAzBhMQsw
CQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cu
ZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBHMzAe
Fw0xMzA4MDExMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJBgNVBAYTAlVTMRUw
EwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5jb20x
IDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IEczMHYwEAYHKoZIzj0CAQYF
K4EEACIDYgAE3afZu4q4C/sLfyHS8L6+c/MzXRq8NOrexpu80JX28MzQC7phW1FG
fp4tn+6OYwwX7Adw9c+ELkCDnOg/QW07rdOkFFk2eJ0DQ+4QE2xy3q6Ip6FrtUPO
Z9wj/wMco+I+o0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjAd
BgNVHQ4EFgQUs9tIpPmhxdiuNkHMEWNpYim8S8YwCgYIKoZIzj0EAwMDaAAwZQIx
AK288mw/EkrRLTnDCgmXc/SINoyIJ7vmiI1Qhadj+Z4y3maTD/HMsQmP3Wyr+mt/
oAIwOWZbwmSNuJ5Q3KjVSaLtx9zRSX8XAbjIho9OjIgrqJqpisXRAL34VOKa5Vt8
sycX
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDxTCCAq2gAwIBAgIQAqxcJmoLQJuPC3nyrkYldzANBgkqhkiG9w0BAQUFADBs
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3

d3cuZGlnaWNlcnQuY29tMSswKQYDVQQDEyJEaWdpQ2VydCBIaWdoIEFzc3VyYW5j
ZSBFViBSb290IENBMB4XDTA2MTExMDAwMDAwMFoXDTMxMTExMDAwMDAwMFowbDEL
MAkGA1UEBhMCVVMxFTATBgNVBAoTDERpZ2lDZXJ0IEluYzEZMBcGA1UECxMQd3d3
LmRpZ2ljZXJ0LmNvbTErMCkGA1UEAxMiRGlnaUNlcnQgSGlnaCBBc3N1cmFuY2Ug
RVYgUm9vdCBDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMbM5XPm
+9S75S0tMqbf5YE/yc0lSbZxKsPVlDRnogocsF9ppkCxxLeyj9CYpKlBWTrT3JTW
PNt0OKRKzE0lgvdKpVMSOO7zSW1xkX5jtqumX8OkhPhPYlG++MXs2ziS4wblCJEM
xChBVfvLWokVfnHoNb9Ncgk9vjo4UFt3MRuNs8ckRZqnrG0AFFoEt7oT61EKmEFB
Ik5lYYeBQVCmeVyJ3hlKV9Uu5l0cUyx+mM0aBhakaHPQNAQTXKFx01p8VdteZOE3
hzBWBOURtCmAEvF5OYiiAhF8J2a3iLd48soKqDirCmTCv2ZdlYTBoSUeh10aUAsg
EsxBu24LUTi4S8sCAwEAAaNjMGEwDgYDVR0PAQH/BAQDAgGGMA8GA1UdEwEB/wQF
MAMBAf8wHQYDVR0OBBYEFLE+w2kD+L9HAdSYJhoIAu9jZCvDMB8GA1UdIwQYMBaA
FLE+w2kD+L9HAdSYJhoIAu9jZCvDMA0GCSqGSIb3DQEBBQUAA4IBAQAcGgaX3Nec
nzyIZgYIVyHbIUf4KmeqvxgydkAQV8GK83rZEWWONfqe/EW1ntlMMUu4kehDLI6z
eM7b41N5cdblIZQB2lWHmiRk9opmzN6cN82oNLFpmyPInngiK3BD41VHMWEZ71jF
hS9OMPagMRYjyOfiZRYzy78aG6A9+MpeizGLYAiJLQwGXFK3xPkKmNEVX58Svnw2
Yzi9RKR/5CYrCsSXaQ3pjOLAEFe4yHYSkVXySGnYvCoCWw9E1CAx2/S6cCZdkGCe
vEsXCS+0yx5DaMkHJ8HSXPfqIbloEpw8nL+e/IBcm2PN7EeqJSdnoDfzAIJ9VNep
+OkuE6N36B9K
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICGTCCAZ+gAwIBAgIQCeCTZaz32ci5PhwLBCou8zAKBggqhkjOPQQDAzBOMQsw
CQYDVQQGEwJVUzEXMBUGA1UEChMORGlnaUNlcnQsIEluYy4xJjAkBgNVBAMTHURp
Z2lDZXJ0IFRMUyBFQ0MgUDM4NCBSb290IEc1MB4XDTIxMDExNTAwMDAwMFoXDTQ2
MDExNDIzNTk1OVowTjELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDkRpZ2lDZXJ0LCBJ
bmMuMSYwJAYDVQQDEx1EaWdpQ2VydCBUTFMgRUNDIFAzODQgUm9vdCBHNTB2MBAG
ByqGSM49AgEGBSuBBAAiA2IABMFEoc8Rl1Ca3iOCNQfN0MsYndLxf3c1TzvdlHJS
7cI7+Oz6e2tYIOyZrsn8aLN1udsJ7MgT9U7GCh1mMEy7H0cKPGEQQil8pQgO4CLp
0zVozptjn4S1mU1YoI71VOeVyaNCMEAwHQYDVR0OBBYEFMFRRVBZqz7nLFr6ICIS
B4CIfBFqMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MAoGCCqGSM49
BAMDA2gAMGUCMQCJao1H5+z8blUD2WdsJk6Dxv3J+ysTvLd6jLRl0mlpYxNjOyZQ
LgGheQaRnUi/wr4CMEfDFXuxoJGZSZOoPHzoRgaLLPIxAJSdYsiJvRmEFOml+wG4
DXZDjC5Ty3zfDBeWUA==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFZjCCA06gAwIBAgIQCPm0eKj6ftpqMzeJ3nzPijANBgkqhkiG9w0BAQwFADBN
MQswCQYDVQQGEwJVUzEXMBUGA1UEChMORGlnaUNlcnQsIEluYy4xJTAjBgNVBAMT
HERpZ2lDZXJ0IFRMUyBSU0E0MDk2IFJvb3QgRzUwHhcNMjEwMTE1MDAwMDAwWhcN
NDYwMTE0MjM1OTU5WjBNMQswCQYDVQQGEwJVUzEXMBUGA1UEChMORGlnaUNlcnQs
IEluYy4xJTAjBgNVBAMTHERpZ2lDZXJ0IFRMUyBSU0E0MDk2IFJvb3QgRzUwggIi
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCz0PTJeRGd/fxmgefM1eS87IE+
ajWOLrfn3q/5B03PMJ3qCQuZvWxX2hhKuHisOjmopkisLnLlvevxGs3npAOpPxG0
2C+JFvuUAT27L/gTBaF4HI4o4EXgg/RZG5Wzrn4DReW+wkL+7vI8toUTmDKdFqgp
wgscONyfMXdcvyej/Cestyu9dJsXLfKB2l2w4SMXPohKEiPQ6s+d3gMXsUJKoBZM

pG2T6T867jp8nVid9E6P/DsjyG244gXazOvswzH016cpVIDPRFtMbzCe88zdH5RD
nU1/cHAN1DrRN/BsnZvAFJNY781BOHW8EwOVfH/jXOnVDdXifBBiqmvwPXbzP6Po
sMH976pXTayGpxi0KcEsDr9kvimM2AItzVwv8n/vFfQMFawKsPHTDU9qTXeXAaDx
Zre3zu/O7Oyldcqs4+Fj97ihBMi8ez9dLRYiVu1ISf6nL3kwJZu6ay0/nTvEF+cd
Lvvyz6b84xQslpghjLSR6Rlgg/IwKwZzUNWYOwbpx4oMYIwo+FKbbuH2TbsGJJvX
KyY//SovcfXWJL5/MZ4PbeiPT02jP/816t9JXkGPhvnxd3lLG7SjXi/7RgLQZhNe
XoVPzthwiHvOAbWWl9fNff2C+MIkwcoBOU+NosEUQB+cZtUMCUbW8tDRSHZWOkPL
tgoRObqME2wGtZ7P6wIDAQABo0IwQDAdBgNVHQ4EFgQUUTMc7TZArxfTJc1paPKv
TiM+s0EwDgYDVR0PAQH/BAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcN
AQEMBQADggIBAGCmr1tfV9qJ20tQqcQjNSH/0GEwhJG3PxDPJY7Jv0Y02cEhJhxw
GXIeo8mH/qlDZJY6yFMECrZBu8RHANmfGBg7sg7zNOok992vIGCukihfNudd5N7H
PNtQOa27PShNlnx2xlv0wdsUpasZYgcYQF+Xkdycx6u1UQ3maVNVzDl92sURVXLF
O4uJ+DQtpBflF+aZfTCIITfNMBc9uPK8qHWgQ9w+iUuQrm0D4ByjoJYJu32jtyoQ
REtGBzRj7TG5BO6jm5qu5jF49OokYTurWGT/u4cnYiWB39yhL/btp/96j1EuMPik
AdKFOV8BmZZvWltwGUb+hmA+rYAQCd05JS9Yf7vSdPD3Rh9GOUrYU9DzLjtxpdRv
/PNn5AeP3SYZ4Y1b+qOTEZvpyDrDVWiakuFSdjjo4bq9+0/V77PnSIMx8IIh47a+
p6tv75/fTM8BuGJqIz3nCU2AG3swpMPdB380vqQmsvZB6Akd4yCYqjdP//fx4ilw
MUc/dNAUFvohigLVigmUdy7yWSiLfFCSCmZ4OIN1xLVaqBHG5cGdZlXPU8Sv13WF
qUITVuwhd4GTWgzqltlJyqEI8pc7bZsEGCREjnwB8twl2F6GmrE52/WRMmrRpnCK
ovfepEWFJqgejF0pW8hL2JpqA15w8oVPbEtoL8pU9ozaMv7Da4M/OMZ+
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFkDCCA3igAwIBAgIQBZsbV56OITLiOQe9p3d1XDANBgkqhkiG9w0BAQwFADBi
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSEwHwYDVQQDExhEaWdpQ2VydCBUcnVzdGVkIFJvb3Qg
RzQwHhcNMTMwODAxMTIwMDAwWhcNMzgwMTE1MTIwMDAwWjBiMQswCQYDVQQGEwJV
UzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQu
Y29tMSEwHwYDVQQDExhEaWdpQ2VydCBUcnVzdGVkIFJvb3QgRzQwggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQC/5pBzaN675F1KPDAiMGkz7MKnJS7JIT3y
ithZwuEppz1Yq3aaza57G4QNxDAf8xukOBbrVsaXbR2rsnnyyhHS5F/WBTxSD1If
xp4VpX6+n6lXFllVcq9ok3DCsrp1mWpzMpTREEQQLt+C8weE5nQ7bXHiLQwb7iDV
ySAdYyktzuxeTsiT+CFhmzTrBcZe7FsavOvJz82sNEBfsXpm7nfISKhmV1efVFiO
DCu3T6cw2Vbuyntd463JT17lNecxy9qTXtyOj4DatpGYQJB5w3jHtrHEtWoYOAMQ
jdjUN6QuBX2I9YI+EJFwq1WCQTLX2wRzKm6RAXwhTNS8rhsDdV14Ztk6MUSaM0C/
CNdaSaTC5qmgZ92kJ7yhTzm1EVgX9yRcRo9k98FpiHaYdj1ZXUJ2h4mXaXpI8OCi
EhtmmnTK3kse5w5jrubU75KSOp493ADkRSWJtppEGSt+wJS00mFt6zPZxd9LBADM
fRyVw4/3IbKyEbe7f/LVjHAsQWCqsWMYRJUadmJ+9oCw++hkpjPRiQfhvbfmQ6QY
uKZ3AeEPlAwhHbJUKSWJbOUOUlFHdL4mrLZBdd56rF+NP8m800EREIvlEFDrMcXK
chYiCd98THU/Y+whX8QgUWtvsauGi0/C1kVfnSD8oR7FwI+isX4KJpn15GkvmB0t
9dmpsh3lGwIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB
hjAdBgNVHQ4EFgQU7NfjgtJxXWRM3y5nP+e6mK4cD08wDQYJKoZIhvcNAQEMBQAD
ggIBALth2X2pbL4XxJEbw6GiAI3jZGgPVs93rnD5/ZpKmbnJeFwMDF/k5hQpVgs2
SV1EY+CtnJYYZhsjDT156W1r1lT40jzBQ0CuHVD1UvyQO7uYmWlrx8GnqGikJ9yd
+SeuMIW59mdNOj6PWTkiU0TryF0Dyu1Qen1iIQqAyHNm0aAFYF/opbSnr6j3bTWc
fFqK1qI4mfN4i/RN0iAL3gTujJtHgXINwBQy7zBZLq7gcfJW5GqXb5JQbZaNaHqa
sjYUegbyJLkJEVDXCLG4iXqEI2FCKeWjzaIgQdfRnGTZ6iahixTXTBmyUEFxPT9N

cCOGDErcgdLMMpSEDQgJlxxPwO5rIHQw0uA5NBCFIRUBCOhVMt5xSdkoF1BN5r5N
0XWs0Mr7QbhDparTwwVETyw2m+L64kW4I1NsBm9nVX9GtUw/bihaeSbSpKhil9Ie
4u1Ki7wb/UdKDd9nZn6yW0HQO+T0O/QEY+nvwlQAUaCKKsnOeMzV6ocEGLPOr0mI
r/OSmbaz5mEP0oUA51Aa5BuVnRmhuZyxm7EAHu/QD09CbMkKvO5D+jpxpchNJqU1
/YldvIViHTLSoCtU7ZpXwdv6EM8Zt4tKG48BtieVU+i2iW1bvGjUI+iLUaJW+fCm
gKDWHrO8Dw9TdSmq6hN35N6MgSGtBxBHEa2HPQfRdbzP82Z+
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIEkTCCA3mgAwIBAgIERWtQVDANBgkqhkiG9w0BAQUFADCBsDELMAkGA1UEBhMC
VVMxFjAUBgNVBAoTDUVudHJ1c3QsIEluYy4xOTA3BgNVBAsTMHd3dy5lbnRydXN0
Lm5ldC9DUFMgaXMgaW5jb3Jwb3JhdGVkIGJ5IHJlZmVyZW5jZTEfMB0GA1UECxMW
KGMpIDIwMDYgRW50cnVzdCwgSW5jLjEtMCsGA1UEAxMkRW50cnVzdCBSb290IENl
cnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTA2MTEyNzIwMjM0MloXDTI2MTEyNzIw
NTM0MlowgbAxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1FbnRydXN0LCBJbmMuMTkw
NwYDVQQLEzB3d3cuZW50cnVzdC5uZXQvQ1BTIGlzIGluY29ycG9yYXRlZCBieSBy
ZWZlcmVuY2UxHzAdBgNVBAsTFihjKSAyMDA2IEVudHJ1c3QsIEluYy4xLTArBgNV
BAMTJEVudHJ1c3QgUm9vdCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBALaVtkNC+sZtKm9I35RMOVcF7sN5EUFo
Nu3s/poBj6E4KPz3EEZmLk0eGrEaTsbRwJWIsMn/MYszA9u3g3s+IIRe7bJWKKf4
4LlAcTfFy0cOlypowCKVYhXbR9n10Cv/gkvJrT7eTNuQgFA/CYqEAOwwCj0Yzfv9
KlmaI5UXLEWeH25DeW0MXJj+SKfFI0dcXv1u5x609mhF0YaDW6KKjbHjKYD+JXGI
rb68j6xSlkuqUY3kEzEZ6E5Nn9uss2rVvDlUccp6en+Q3X0dgNmBu1kmwhH+5pPi
94DkZfs0Nw4pgHBNrziGLp5/V6+eF67rHMsoIV+2HNjnogQi+dPa2MsCAwEAAaOB
sDCBrTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zArBgNVHRAEJDAi
gA8yMDA2MTEyNzIwMjM0MlqBDzIwMjYxMTI3MjA1MzQyWjAfBgNVHSMEGDAWgBRo
kORnpKZTgMeGZqTx90tD+4S9bTAdBgNVHQ4EFgQUaJDkZ6SmU4DHhmak8fdLQ/uE
vW0wHQYJKoZIhvZ9B0EABBAwDhsIVjcuMTo0LjADAgSQMA0GCSqGSIb3DQEBBQUA
A4IBAQCT1DCw1wMgKtD5Y+iRDAUgqV8ZyntyTtSx29CW+1RaGSwMCPeyvIWonX9t
O1KzKtvn1ISMY/YPyyYBkVBs9F8U4pN0wBOeMDpQ47RgxRzwIkSNcUesyBrJ6Zua
AGAT/3B+XxFNSRuzFVJ7yVTav52Vr2ua2J7p8eRDjeIRRDq/r72DQnNSi6q7pynP
9WQcCk3RvKqsnyrQ/39/2n3qse0wJcGE2jTSW3iDVuycNsMm4hH2Z0kdkquM++v/
eu6FSqdQgPCnXEqULl8FmTxSQeDNtGPPAUO6nIPcj2A781q0tHuu2guQOHXvgR1m
0vdXcDazv/wor3ElhVsT/h5/WrQ8
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIC+TCCAoCgAwIBAgINAKaLeSkAAAAAUNCR+TAKBggqhkjOPQQDAzCBvzELMAkG
A1UEBhMCVVMxFjAUBgNVBAoTDUVudHJ1c3QsIEluYy4xKDAmBgNVBAsTH1NlZSB3
d3cuZW50cnVzdC5uZXQvbGVnYWwtdGVybXMxOTA3BgNVBAsTMChjKSAyMDEyIEVu
dHJ1c3QsIEluYy4gLSBmb3IgYXV0aG9yaXplZCB1c2Ugb25seTEzMDEGA1UEAxMq
RW50cnVzdCBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5IC0gRUMxMB4XDTEy
MTIxODE1MjUzNloXDTM3MTIxODE1NTUzNlowgb8xCzAJBgNVBAYTAlVTMRYwFAYD
VQQKEw1FbnRydXN0LCBJbmMuMSgwJgYDVQQLEx9TZWUgd3d3LmVudHJ1c3QubmV0
L2xlZ2FsLXRlcm1zMTkwNwYDVQQLEzAoYykgMjAxMiBFbnRydXN0LCBJbmMuIC0g
Zm9yIGF1dGhvcml6ZWQgdXNlIG9ubHkxMzAxBgNVBAMTKkVudHJ1c3QgUm9vdCBD

ZXJ0aWZpY2F0aW9uIEF1dGhvcml0eSAtIEVDMTB2MBAGByqGSM49AgEGBSuBBAAi
A2IABIQTydC6bUF74mzQ61VfZgIaJPRbiWlH47jCffHyAsWfoPZb1YsGGYZPUxBt
ByQnoaD41UcZYUx9ypMn6nQM72+WCf5j7HBdNq1nd67JnXxVRDqiY1Ef9eNi1KlH
Bz7MIKNCMEAwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0O
BBYEFLdj5xrdjekIplWDpOBqUEFlEUJJMAoGCCqGSM49BAMDA2cAMGQCMGF52OVC
R98crlOZF7ZvHH3hvxGU0QOIdeSNiaSKd0bebWHvAvX7td/M/k7//qnmpwIwW5nX
hTcGtXsI/esni0qU+eH6p44mCOh8kmhtc9hvJqwhAriZtyZBWyVgrtBIGu4G
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIEPjCCAyagAwIBAgIESlOMKDANBgkqhkiG9w0BAQsFADCBvjELMAkGA1UEBhMC
VVMxFjAUBgNVBAoTDUVudHJ1c3QsIEluYy4xKDAmBgNVBAsTH1NlZSB3d3cuZW50
cnVzdC5uZXQvbGVnYWwtdGVybXMxOTA3BgNVBAsTMChjKSAyMDA5IEVudHJ1c3Qs
IEluYy4gLSBmb3IgYXV0aG9yaXplZCB1c2Ugb25seTEyMDAGA1UEAxMpRW50cnVz
dCBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5IC0gRzIwHhcNMDkwNzA3MTcy
NTU0WhcNMzAxMjA3MTc1NTU0WjCBvjELMAkGA1UEBhMCVVMxFjAUBgNVBAoTDUVu
dHJ1c3QsIEluYy4xKDAmBgNVBAsTH1NlZSB3d3cuZW50cnVzdC5uZXQvbGVnYWwt
dGVybXMxOTA3BgNVBAsTMChjKSAyMDA5IEVudHJ1c3QsIEluYy4gLSBmb3IgYXV0
aG9yaXplZCB1c2Ugb25seTEyMDAGA1UEAxMpRW50cnVzdCBSb290IENlcnRpZmlj
YXRpb24gQXV0aG9yaXR5IC0gRzIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC6hLZy254Ma+KZ6TABp3bqMriVQRrJ2mFOWHLP/vaCeb9zYQYKpSfYs1/T
RU4cctZOMvJyig/3gxnQaoCAAEUesMfnmr8SVycco2gvCoe9amsOXmXzHHfV1IWN
cCG0szLni6LVhjkCsbjSR87kyUnEO6fe+1R9V77w6G7CebI6C1XiUJgWMhNcL3hW
wcKUs/Ja5CeanyTXxuzQmyWC48zCxEXFjJd6BmsqEZ+pCm5IO2/b1BEZQvePB7/1
U1+cPvQXLOZprE4yTGJ36rfo5bs0vBmLrpxR57d+tVOxMyLlbc9wPBr64ptntoP0
jaWvYkxN4FisZDQSA/i2jZRjJKRxAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAP
BgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBRqciZ60B7vfec7aVHUbI2fkBJmqzAN
BgkqhkiG9w0BAQsFAAOCAQEAeZ8dlsa2eT8ijYfThwMEYGprmi5ZiXMRrEPR9RP/
jTkrwPK9T3CMqS/qF8QLVJ7UG5aYMzyorWKiAHarWWluBh1+xLlEjZivEtRh2woZ
Rkfz6/djwUAFQKXSt/S1mja/qYh2iARVBCuch38aNzx+LaUa2NSJXsq9rD1s2G2v
1fN2D807iDginWyTmsQ9v4IbZT+mD12q/OWyFcq1rca8PdCE6OoGcrBNOTJ4vz4R
nAuknZoh8/CbCzB428Hch0P+vGOaysXCHMnHjf87ElgI5rY97HosTvuDls4MPGmH
VHOkc8KT/1EQrBVUAdj8BbGJoX90g5pJ19xOe4pIb4tF9g==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICejCCAgCgAwIBAgIQMZch7a+JQn81QYehZ1ZMbTAKBggqhkjOPQQDAzBuMQsw
CQYDVQQGEwJFUzEcMBoGA1UECgwTRmlybWFwcm9mZXNpb25hbCBTQTEYMBYGA1UE
YQwPVkFURVMtQTYyNjM0MDY4MScwJQYDVQQDDB5GSVJNQVBST0ZFU0lPTkFMIENB
IFJPT1QtQSBXRUIwHhcNMjIwNDA2MDkwMTM2WhcNNDcwMzMxMDkwMTM2WjBuMQsw
CQYDVQQGEwJFUzEcMBoGA1UECgwTRmlybWFwcm9mZXNpb25hbCBTQTEYMBYGA1UE
YQwPVkFURVMtQTYyNjM0MDY4MScwJQYDVQQDDB5GSVJNQVBST0ZFU0lPTkFMIENB
IFJPT1QtQSBXRUIwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAARHU+osEaR3xyrq89Zf
e9MEkVz6iMYiuYMQYneEMy3pA4jU4DP37XcsSmDq5G+tbbT4TIqk5B/K6k84Si6C
cyvHZpsKjECcfIr28jlgst7L7Ljkb+qbXbdTkBgyVcUgt5SjYzBhMA8GA1UdEwEB
/wQFMAMBAf8wHwYDVR0jBBgwFoAUk+FDY1w8ndYn81LsF7Kpryz3dvgwHQYDVR0O

BBYEFJPhQ2NcPJ3WJ/NS7Beyqa8s93b4MA4GA1UdDwEB/wQEAwIBBjAKBggqhkjO
PQQDAwNoADBlAjAdfKR7w4l1M+E7qUW/Runpod3JIha3RxEL2Jq68cgLcFBTApFw
hVmpHqTm6iMxoAACMQD94vizrxa5HnPEluPBMBnYfubDl94cT7iJLzPrSA8Z94dG
XSaQpYXFuXqUPoeovQA=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFiDCCA3CgAwIBAgIIfQmX/vBH6nowDQYJKoZIhvcNAQELBQAwYjELMAkGA1UE
BhMCQ04xMjAwBgNVBAoMKUdVQU5HIERPTkcgQ0VSVElGSUNBVEUgQVVUSE9SSVRZ
IENPLixMVEQuMR8wHQYDVQQDDBZHRENBIFRydXN0QVVUSCBSNSBST09UMB4XDTE0
MTEyNjA1MTMxNVoXDTQwMTIzMTE1NTk1OVowYjELMAkGA1UEBhMCQ04xMjAwBgNV
BAoMKUdVQU5HIERPTkcgQ0VSVElGSUNBVEUgQVVUSE9SSVRZIENPLixMVEQuMR8w
HQYDVQQDDBZHRENBIFRydXN0QVVUSCBSNSBST09UMIICIjANBgkqhkiG9w0BAQEF
AAOCAg8AMIICCgKCAgEA2aMW8Mh0dHeb7zMNOwZ+Vfy1YI92hhJCfVZmPoiC7XJj
Dp6L3TQsAlFRwxn9WVSEyfFrs0yw6ehGXTjGoqcuEVe6ghWinI9tsJlKCvLriXBj
TnnEt1u9ol2x8kECK62pOqPseQrsXzrj/e+APK00mxqriCZ7VqKChh/rNYmDf1+u
KU49tm7srsHwJ5uu4/Ts765/94Y9cnrrpftZTqfrlYwiOXnhLQiPzLyRuEH3FMEj
qcOtmkVEs7LXLM3GKeJQEK5cy4KOFxg2fZfmiJqwTTQJ9Cy5WmYqsBebnh52nUpm
MUHfP/vFBu8btn4aRjb3ZGM74zkYI+dndRTVdVeSN72+ahsmUPI2JgaQxXABZG12
ZuGR224HwGGALrIuL4xwp9E7PLOR5G62xDtw8mySlwnNR30YwPO7ng/Wi64HtloP
zgsMR6flPri9fcebNaBhlzpBdRfMK5Z3KpIhHtmVdiBnaM8Nvd/WHwlqmuLMc3Gk
L30SgLdTMEZeS1SZD2fJpcjyIMGC7J0R38IC+xo70e0gmu9lZJIQDSri3nDxGGeC
jGHeuLzRL5z7D9Ar7Rt2ueQ5Vfj4oR24qoAATILnsn8JuLwwoC8N9VKejveSswoA
HQBUlwbgsQfZxw9cZX08bVlX5O2ljelAU58VS6Bx9hoh49pwBiFYFIeFd3mqgnkC
AwEAAaNCMEAwHQYDVR0OBBYEFOLJQJ9NzuiaoXzPDj9lxSmIahlRMA8GA1UdEwEB
/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3DQEBCwUAA4ICAQDRSVfg
p8xoWLoBDysZzY2wYUWsEe1jUGn4H3++Fo/9nesLqjJHdtJnJO29fDMylyrHBYZm
DRd9FBUb1Ov9H5r2XpdptxolpAqzkT9fNqyL7FeoPueBihhXOYV0GkLH6VsTX4/5
COmSdI31R9KrO9b7eGZONn356ZLpBN79SWP8bfsUcZNnL0dKt7n/HipzcEYwv1ry
L3ml4Y0M2fmyYzeMN2WFcGpcWwlyua1jPLHd+PwyvzeG5LuOmCd+uh8W4XAR8gPf
JWIyJyYYMoSf/wA6E7qaTfRPuBRwIrHKK5DOKcFw9C+df/KQHtZa37dG/OaG+svg
IHZ6uqbL9XzeYqWxi+7egmaKTjowHz+Ay60nugxe19CxVsp3cbK1daFQqUBDF8Io
2c9Si1vIY9RCPqAzekYu9wogRlR+ak8x8YF+QnQ4ZXMn7sZ8uI7XpTrXmKGcjBBV
09tL7ECQ8s1uV9JiDnxXk7Gnbc2dg7sq5+W2O3FYrf3RRbxake5TFW/TRQl1brqQ
XR4EzzffHqhmsYzmIGrv/EhOdJhCrylvLmrH+33RZjEizIYAfmaDDEL0vTSSwxrq
T8p+ck0LcIymSLumoRT2+1hEmRSuqguTaaApJUqlyyvdimYHFngVV3Eb7PVHhPOe
MTd61X8kreS8/f3MboPoDKi3QWwH3b08hpcv0g==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFgjCCA2qgAwIBAgILWku9WvtPilv6ZeUwDQYJKoZIhvcNAQELBQAwTTELMAkG
A1UEBhMCQVQxIzAhBgNVBAoTGmUtY29tbWVyY2UgbW9uaXRvcmluZyBHbWJIMRkw
FwYDVQQDExBHTE9CQUxUUlVTVCAyMDIwMB4XDTIwMDIxMDAwMDAwMFoXDTQwMDYx
MDAwMDAwMFowTTELMAkGA1UEBhMCQVQxIzAhBgNVBAoTGmUtY29tbWVyY2UgbW9u
aXRvcmluZyBHbWJIMRkwFwYDVQQDExBHTE9CQUxUUlVTVCAyMDIwMIICIjANBgkq
hkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAri5WrRsc7/aVj6B3GyvTY4+ETUWiD59b

RatZe1E0+eyLinjF3WuvvcTfk0Uev5E4C64OFudBc/jbu9G4UeDLgztzOG53ig9Z
YybNpyrOVPu44sB8R85gfD+yc/LAGbaKkoc1DZAoouQVBGM+uq/ufF7MpotQsjj3
QWPKzv9pj2gOlTblzLmMCcpL3TGQlsjMH/1WljTbjhzqLL6FLmPdqqmV0/0plRPw
yJiT2S0WR5ARg6I6IqIoV6Lr/sCMKKCmfecqQjuCgGOlYx8ZzHyyZqjC0203b+J+
BlHZRYQfEs4kUmSFC0iATоexIiIwquuuvuAC4EDosEKAA1GqtH6qRNdDYfOiaxaJ
SaSjpCuKAsR49GiKweR6NrFvG5Ybd0mN1MkGco/PU+PcF4UgStyYJ9ORJitHHmkH
r96i5OTUawuzXnzUJIBHKWk7buis/UDr2O1xcSvy6Fgd60GXIsUf1DnQJ4+H4xj0
4KlGDfV0OoIu0G4skaMxXDtG6nsEEFZegB31pWXogvziB4xiRfUg3kZwhqG8k9Me
dKZssCz3AwyIDMvUclOGvGBG85hqwvG/Q/lwIHfKN0F5VVJjjVsSn8VoxIidrPIw
q7ejMZdnrY8XD2zHc+0klGvIg5rQmjdJBKuxFshsSUktq6HQjJLyQUp5ISXbY9e2
nKd+Qmn7OmMCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AQYwHQYDVR0OBBYEFNwuH9FhN3nkq9XVsxJxaD1qaJwiMB8GA1UdIwQYMBaAFNwu
H9FhN3nkq9XVsxJxaD1qaJwiMA0GCSqGSIb3DQEBCwUAA4ICAQCR8EICaEDuw2jA
VC/f7GLDw56KoDEoqoOOpFaWEhCGVrqXctJUMHytGdUdaG/7FELYjQ7ztdGl4wJC
XtzoRlgHNQIw4Lx0SsFDKv/bGtCwr2zD/cuz9X9tAy5ZVp0tLTWMstZDFyySCstd
6IwPS3BD0IL/qMy/pJTAvoe9iuOTe8aPmxadJ2W8esVCgmxcB9CpwYhgROmYhRZf
+I/KARDOJcP5YBugxZfD0yyIMaK9MOzQ0MAS8cE54+X1+NZK3TTN+2/BT+MAi1bi
kvcoskJ3ciNnxz8RFbLEAwW+uxF7Cr+obuf/WEPPm2eggAe2HcqtbepBEX4tdJP7
wry+UUTF72glJ4DjyKDUEuzZpTcdN3y0kcra1LGWge9oXHYQSa9+pTeAsRxSvTOB
TI/53WXZFM2KJVj04sWDpQmQ1GwUY7VA3+vA/MRYfg0UFodUJ25W5HCEuGwyEn6C
MUO+1918oa2u1qsgEu8KwxCMSZY13At1XrFP1U80DhEgB3VDRemjEdqso5nCtnkn
4rnvyOL2NSl6dPrFf4IFYqYK6miyeUcGbvJXqBUzxvd4Sj1Ce2t+/vdG6tHrju+I
aFvowdlxfv1k7/9nR4hYJS8+hge9+6jlgqispdNpQ80xiEmEU5LAsTkbOYMBMMTy
qfrQA71yN2BWHzZ8vTmR9W0Nv3vXkg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFVzCCAz+gAwIBAgINAgPlk28xsBNJiGuiFzANBgkqhkiG9w0BAQwFADBHMQsw
CQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2xlIFRydXN0IFNlcnZpY2VzIExMQzEU
MBIGA1UEAxMLR1RTIFJvb3QgUjEwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAw
MDAwWjBHMQswCQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2xlIFRydXN0IFNlcnZp
Y2VzIExMQzEUMBIGA1UEAxMLR1RTIFJvb3QgUjEwggIiMA0GCSqGSIb3DQEBAQUA
A4ICDwAwggIKAoICAQC2EQKLHuOhd5s73L+UPreVp0A8of2C+X0yBoJx9vaMf/vo
27xqLpeXo4xL+Sv2sfnOhB2x+cWX3u+58qPpvBKJXqeqUqv4IyfLpLGcY9vXmX7w
Cl7raKb0xlpHDU0QM+NOsROjyBhsS+z8CZDfnWQpJSMHobTSPS5g4M/SCYe7zUjw
TcLCeoiKu7rPWRnWr4+wB7CeMfGCwcDfLqZtbBkOtdh+JhpFAz2weaSUKK0Pfybl
qAj+lug8aJRT7oM6iCsVlgmy4HqMLnXWnOunVmSPlk9orj2XwoSPwLxAwAtcvfaH
szVsrBhQf4TgTM2S0yDpM7xSma8ytSmzJSq0SPly4cpk9+aCEI3oncKKiPo4Zor8
Y/kB+Xj9e1x3+naH+uzfsQ55lVe0vSbv1gHR6xYKu44LtcXFilWr06zqkUspzBmk
MiVOKvFlRNACzqrOSbTqn3yDsEB750Orp2yjj32JgfpMpf/VjsPOS+C12LOORc92
wO1AK/1TD7Cn1TsNsYqiA94xrcx36m97PtbfkSIS5r762DL8EGMUUXLeXdYWk70p
aDPvOmbsB4om3xPXV2V4J95eSRQAogB/mqghtqmxlbCluQ0WEdrHbEg8QOB+DVrN
VjzRlwW5y0vtOUucxD/SVRNuJLDWcfr0wbrM7Rv1/oFB2ACYPTrIrnqYNxgFlQID
AQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4E
FgQU5K8rJnEaK0gnhS9SZizv8IkTcT4wDQYJKoZIhvcNAQEMBQADggIBAJ+qQibb
C5u+/x6Wki4+omVKapi6Ist9wTrYggoGxval3sBOh2Z5ofmmWJyq+bXmYOfg6LEe
QkEzCzc9zolwFcq1JKjPa7XSQCGYzyI0zzvFIoTgxQ6KfF2I5DUkzps+GlQebtuy

h6f88/qBVRRiClmpIgUxPoLW7ttXNLwzldMXG+gnoot7TiYaelpkttGsN/H9oPM4
7HLwEXWdyzRSjeZ2axfG34arJ45JK3VmgRAhpuo+9K4l/3wV3s6MJT/KYnAK9y8J
ZgfIPxz88NtFMN9iiMG1D53Dn0reWVlHxYciNuaCp+0KueIHoI17eko8cdLiA6Ef
MgfdG+RCzgwARWGAtQsgWSl4vflVy2PFPEz0tv/bal8xa5meLMFrUKTX5hgUvYU/
Z6tGn6D/Qqc6f1zLXbBwHSs09dR2CQzreExZBfMzQsNhFRAbd03OIozUhfJFfbdT
6u9AWpQKXCBfTkBdYiJ23//OYb2MI3jSNwLgjt7RETeJ9r/tSQdirpLsQBqvFAnZ
0E6yove+7u7Y/9waLd64NnHi/Hm3lCXRSHNboTXns5lndcEZOitHTtNCjv0xyBZm
2tIMPNuzjsmhDYAPexZ3FL//2wmUspO8IFgV6dtxQ/PeEMMA3KgqlbbC1j+Qa3bb
bP6MvPJwNQzcmRk13NfIRmPVNnGuV/u3gm3c
-----END
CERTIFICATE-----

-----BEGIN
CERTIFICATE-----
MIIFVzCCAz+gAwIBAgINAgPlrsWNBCUaqxElqjANBgkqhkiG9w0BAQwFADBHMQsw
CQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2xlIFRydXN0IFNlcnZpY2VzIExMQzEU
MBIGA1UEAxMLR1RTIFJvb3QgUjIwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAw
MDAwWjBHMQswCQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2xlIFRydXN0IFNlcnZp
Y2VzIExMQzEUMBIGA1UEAxMLR1RTIFJvb3QgUjIwggIiMA0GCSqGSIb3DQEBAQUA
A4ICDwAwggIKAoICAQDO3v2m++zsFDQ8BwZabFn3GTXd98GdVarTzTukk3LvCvpt
nfbwhYBboUhSnznFt+4orO/LdmgUud+tAWyZH8QiHZ/+cnfgLFuv5AS/T3KgGjSY
6Dlo7JUle3ah5mm5hRm9iYz+re026nO8/4Piy33B0s5Ks40FnotJk9/BW9BuXvAu
MC6C/Pq8tBcKSOWIm8Wba96wyrQD8Nr0kLhlZPdcTK3ofmZemde4wj7I0BOdre7k
RXuJVfeKH2JShBKzwkCX44ofR5GmdFrS+LFjKBC4swm4VndAoiaYecb+3yXuPuWg
f9RhD1FLPD+M2uFwdNjCaKH5wQzpoeJ/u1U8dgbuak7MkogwTZq9TwtImoS1mKPV
+3PBV2HdKFZ1E66HjucMUQkQdYhMvI35ezzUIkgfKtzra7tEscszcTJGr61K8Yzo
dDqs5xoic4DSMPclQsciOzsSrZYuxsN2B6ogtzVJV+mSSeh2FnIxZyuWfoqjx5RW
Ir9qS34BIbIjMt/kmkRtWVtd9QCgHJvGeJeNkP+byKq0rxFROV7Z+2et1VsRnTKa
G73VululycslaVNVJ1zgyjbLiGH7HrfQy+4W+9OmTN6SpdTi3/UGVN4unUu0kzCq
gc7dGtxRcw1PcOnlthYhGXmy5okLdWTK1au8CcEYof/UVKGFPP0UJAOyh9OktwID
AQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4E
FgQUu//KjiOfT5nK2+JopqUVJxce2Q4wDQYJKoZIhvcNAQEMBQADggIBAB/Kzt3H
vqGf2SdMC9wXmBFqiN495nFWcrKeGk6c1SuYJF2ba3uwM4IJvd8lRuqYnrYb/oM8
0mJhwQTtzuDFycgTE1XnqGOtjHsB/ncw4c5omwX4Eu55MaBBRTUoCnGkJE+M3DyC
B19m3H0Q/gxhswWV7uGugQ+o+MePTagjAiZrHYNSVc61LwDKgEDg4XSsYPWHgJ2u
NmSRXbBoGOqKYcl3qJfEycel/FVL8/B/uWU9J2jQzGv6U53hkRrJXRqWbTKH7QMg
yALOWr7Z6v2yTcQvG99fevX4i8buMTolUVVnjWQye+mew4K6Ki3pHrTgSAai/Gev
HyICc/sgCq+dVEuhzf9gR7A/Xe8bVr2XIZYtCtFenTgCR2y59PYjJbigapordwj6
xLEokCZYCDzifqrXPW+6MYgKBesntaFJ7qBFVHvmJ2WZICGoo7z7GJa7Um8M7YNR
TOlZ4iBgxcJlkoKM8xAfDoqXvneCbT+PHV28SSe9zE8P4c52hgQjxcCMElv924Sg
JPFI/2R80L5cFtHvma3AH/vLrrw4IgYmZNralw4/KBVEqE8AyvCazM90arQ+POuV
7LXTWtiBmelDGDfrs7vRWGJB82bSj6p4lVQgw1oudCvV0b4YacCs1aTPObpRhANl
6WLAYv7YTVWW4tAR+kg0Eeye7QUd5MjWHYbL
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICCTCCAY6gAwIBAgINAgPluILrIPglJ209ZjAKBggqhkjOPQQDAzBHMQswCQYD

VQQGEwJVUzEiMCAGA1UEChMZR29vZ2xlIFRydXN0IFNlcnZpY2VzIExMQzEUMBIG
A1UEAxMLR1RTIFJvb3QgUjMwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAwMDAw
WjBHMQswCQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2xlIFRydXN0IFNlcnZpY2Vz
IExMQzEUMBIGA1UEAxMLR1RTIFJvb3QgUjMwdjAQBgcqhkjOPQIBBgUrgQQAIgNi
AAQfTzOHMymKoYTey8chWEGJ6ladK0uFxh1MJ7x/JlFyb+Kf1qPKzEUURout736G
jOyxfi//qXGdGIRFBEFVbivqJn+7kAHjSxm65FSWRQmx1WyRRK2EE46ajA2ADDL2
4CejQjBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBTB8Sa6oC2uhYHP0/EqEr24Cmf9vDAKBggqhkjOPQQDAwNpADBmAjEA9uEglRR7
VKOQFhG/hMjqb2sXnh5GmCCbn9MN2azTL818+FsuVbu/3ZL3pAzcMeGiAjEA/Jdm
ZuVDFhOD3cffL74UOO0BzrEXGhF16b0DjyZ+hOXJYKaV11RZt+cRLInUue4X
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICCTCCAY6gAwIBAgINAgPlwGjvYxqccpBQUjAKBggqhkjOPQQDAzBHMQswCQYD
VQQGEwJVUzEiMCAGA1UEChMZR29vZ2xlIFRydXN0IFNlcnZpY2VzIExMQzEUMBIG
A1UEAxMLR1RTIFJvb3QgUjQwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAwMDAw
WjBHMQswCQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2xlIFRydXN0IFNlcnZpY2Vz
IExMQzEUMBIGA1UEAxMLR1RTIFJvb3QgUjQwdjAQBgcqhkjOPQIBBgUrgQQAIgNi
AATzdHOnaItgrkO4NcWBMHtLSZ37wWHO5t5GvWvVYRg1rkDdc/eJkTBa6zzuhXyi
QHY7qca4R9gq55KRanPpsXI5nymfopjTX15YhmUPoYRlBtHci8nHc8iMai/lxKvR
HYqjQjBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBSATNbrdP9JNqPV2Py1PsVq8JQdjDAKBggqhkjOPQQDAwNpADBmAjEA6ED/g94D
9J+uHXqnLrmvT/aDHQ4thQEd0dlq7A/Cr8deVl5c1RxYIigL9zC2L7F8AjEA8GE8
p/SgguMh1YQdc4acLa/KNJvxn7kjNuK8YAOdgLOaVsjh4rsUecrNIdSUtUlD
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIB3DCCAYOgAwIBAgINAgPlfvU/k/2lCSGypjAKBggqhkjOPQQDAjBQMSQwIgYD
VQQLExtHbG9iYWxTaWduIEVDQyBSb290IENBIC0gUjQxEzARBgNVBAoTCkdsb2Jh
bFNpZ24xEzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMTIxMTEzMDAwMDAwWhcNMzgw
MTE5MDMxNDA3WjBQMSQwIgYDVQQLExtHbG9iYWxTaWduIEVDQyBSb290IENBIC0g
UjQxEzARBgNVBAoTCkdsb2JhbFNpZ24xEzARBgNVBAMTCkdsb2JhbFNpZ24wWTAT
BgcqhkjOPQIBBggqhkjOPQMBBwNCAAS4xnnTj2wlDp8uORkcA6SumuU5BwkWymOx
uYb4ilfBV85C+nOh92VC/x7BALJucw7/xyHlGKSq2XE/qNS5zowdo0IwQDAOBgNV
HQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUVLB7rUW44kB/
+wpu+74zyTyjhNUwCgYIKoZIzj0EAwIDRwAwRAIgIk90crlgr/HmnKAWBVBfw147
bmF0774BxL4YSFlhgjICICadVGNA3jdgUM/I2O2dgq43mLyjj0xMqTQrbO/7lZsm
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICHjCCAaSgAwIBAgIRYFlJ4CYuu1X5CneKcflK2GwwCgYIKoZIzj0EAwMwUDEk
MCIGA1UECxMbR2xvYmFsU2lnbiBFQ0MgUm9vdCBDQSAtIFI1MRMwEQYDVQQKEwpH
bG9iYWxTaWduMRMwEQYDVQQDEwpHbG9iYWxTaWduMB4XDTEyMTExMzAwMDAwMFoX
DTM4MDExOTAzMTQwN1owUDEkMCIGA1UECxMbR2xvYmFsU2lnbiBFQ0MgUm9vdCBD
QSAtIFI1MRMwEQYDVQQKEwpHbG9iYWxTaWduMRMwEQYDVQQDEwpHbG9iYWxTaWdu

MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAER0UOlvt9Xb/pOdEh+J8LttV7HpI6SFkc
8GIxLcB6KP4ap1yztsyX50XUWPrRd21DosCHZTQKH3rd6zwzocWdTaRvQZU4f8ke
hOvRnkmSh5SHDDqFSmafnVmTTZdhBoZKo0IwQDAOBgNVHQ8BAf8EBAMCAQYwDwYD
VR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUPeYpSJvqB8ohREom3m7e0oPQn1kwCgYI
KoZIzj0EAwMDaAAwZQIxAOVpEslu28YxuglB4Zf4+/2a4n0Sye18ZNPLBSWLVtmg
515dTguDnFt2KaAJJiFqYgIwcdK1j1zqO+F4CYWodZI7yFz9SO8NdCKoCOJuxUnO
xwy8p2Fp8fc74SrL+SvzZpA3
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDXzCCAkegAwIBAgILBAAAAAABIVhTCKIwDQYJKoZIhvcNAQELBQAwTDEgMB4G
A1UECxMXR2xvYmFsU2lnbiBSb290IENBIC0gUjMxEzARBgNVBAoTCkdsb2JhbFNp
Z24xEzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMDkwMzE4MTAwMDAwWhcNMjkwMzE4
MTAwMDAwWjBMMSAwHgYDVQQLExdHbG9iYWxTaWduIFJvb3QgQ0EgLSBSMzETMBEG
A1UEChMKR2xvYmFsU2lnbjETMBEGA1UEAxMKR2xvYmFsU2lnbjCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAMwldpB5BngiFvXAg7aEyiie/QV2EcWtiHL8
RgJDx7KKnQRfJMsuS+FggkbhUqsMgUdwbN1k0ev1LKMPgj0MK66X17YUhhB5uzsT
gHeMCOFJ0mpiLx9e+pZo34knlTifBtc+ycsmWQ1z3rDI6SYOgxXG71uL0gRgykmm
KPZpO/bLyCiR5Z2KYVc3rHQU3HTgOu5yLy6c+9C7v/U9AOEGM+iCK65TpjoWc4zd
QQ4gOsC0p6Hpsk+QLjJg6VfLuQSSaGjlOCZgdbKfd/+RFO+uIEn8rUAVSNECMWEZ
XriX7613t2Saer9fwRPvm2L7DWzgVGkWqQPabumDk3F2xmmFghcCAwEAAaNCMEAw
DgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFI/wS3+o
LkUkrk1Q+mOai97i3Ru8MA0GCSqGSIb3DQEBCwUAA4IBAQBLQNvAUKr+yAzv95ZU
RUm7lgAJQayzE4aGKAczymvmdLm6AC2upArT9fHxD4q/c2dKg8dEe3jgr25sbwMp
jjM5RcOO5LlXbKr8EpbsU8Yt5CRsuZRj+9xTaGdWPoO4zzUhw8lo/s7awlOqzJCK
6fBdRoyV3XpYKBovHd7NADdBj+1EbddTKJd+82cEHhXXipa0095MJ6RMG3NzdvQX
mcIfeg7jLQitChws/zyrVQ4PkX4268NXSb7hLi18YIvDQVETI53O9zJrlAGomecs
Mx86OyXShkDOOyyGeMlhLxS67ttVb9+E7gUJTb0o2HLO02JQZR7rkpeDMdmztcpH
WD9f
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFgzCCA2ugAwIBAgIORea7A4Mzw4VlSOb/RVEwDQYJKoZIhvcNAQEMBQAwTDEg
MB4GA1UECxMXR2xvYmFsU2lnbiBSb290IENBIC0gUjYxEzARBgNVBAoTCkdsb2Jh
bFNpZ24xEzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMTQxMjEwMDAwMDAwWhcNMzQx
MjEwMDAwMDAwWjBMMSAwHgYDVQQLExdHbG9iYWxTaWduIFJvb3QgQ0EgLSBSNjET
MBEGA1UEChMKR2xvYmFsU2lnbjETMBEGA1UEAxMKR2xvYmFsU2lnbjCCAiIwDQYJ
KoZIhvcNAQEBBQADggIPADCCAgoCggIBAJUH6HPKZvnsFMp7PPcNCPG0RQssgrRI
xutbPK6DuEGSMxSkb3/pKszGsIhrxbaJ0cay/xTOURQh7ErdG1rG1ofuTToVBu1k
ZguSgMpE3nOUTvOniX9PeGMIyBJQbUJmL025eShNUhqKGoC3GYEOfsSKvGRMIRxD
aNc9PIrFsmbVkJq3MQbFvuJtMgamHvm566qjuL++gmNQ0PAYid/kD3n16qIfKtJw
LnvnvJO7bVPiSHyMEAc4/2ayd2F+4OqMPKq0pPbzlUoSB239jLKJz9CgYXfIWHSw
1CM69106yqLbnQneXUQtkPGBzVeS+n68UARjNN9rkxi+azayOeSsJDa38O+2HBNX
k7besvjihbdzorg1qkXy4J02oW9UivFyVm4uiMVRQkQVlO6jxTiWm05OWgtH8wY2
SXcwvHE35absIQh1/OZhFj931dmRl4QKbNQCTXTAFO39OfuD8l4UoQSwC+n+7o/h
bguyCLNhZglqsQY6ZZZZwPA1/cnaKI0aEYdwgQqomnUdnjqGBQCe24DWJfncBZ4n
WUx2OVvq+aWh2IMP0f/fMBH5hc8zSPXKbWQULHpYT9NLCEnFlWQaYw55PfWzjMpY

rZxCRXluDocZXFSxZba/jJvcE+kNb7gu3GduyYsRtYQUigAZcIN5kZeR1Bonvzce
MgfYFGM8KEyvAgMBAAGjYzBhMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTAD
AQH/MB0GA1UdDgQWBBSubAWjkxPioufi1xzWx/B/yGdToDAfBgNVHSMEGDAWgBSu
bAWjkxPioufi1xzWx/B/yGdToDANBgkqhkiG9w0BAQwFAAOCAgEAgyXt6NH9lVLN
nsAEoJFp5lzQhN7craJP6Ed41mWYqVuoPId8AorRbrcWc+ZfwFSY1XS+wc3iEZGt
Ixg93eFyRJa0lV7Ae46ZeBZDE1ZXs6KzO7V33EByrKPrmzU+sQghoefEQzd5Mr61
55wsTLxDKZmOMNOsIeDjHfrYBzN2VAAiKrlNIC5waNrlU/yDXNOd8v9EDERm8tLj
vUYAGm0CuiVdjaExUd1URhxN25mW7xocBFymFe944Hn+Xds+qkxV/ZoVqW/hpvvf
cDDpw+5CRu3CkwWJ+n1jez/QcYF8AOiYrg54NMMl+68KnyBr3TsTjxKM4kEaSHpz
oHdpx7Zcf4LIHv5YGygrqGytXm3ABdJ7t+uA/iU3/gKbaKxCXcPu9czc8FB10jZp
nOZ7BN9uBmm23goJSFmH63sUYHpkqmlD75HHTOwY3WzvUy2MmeFe8nI+z1TIvWfs
pA9MRf/TuTAjB0yPEL+GltmZWrSZVxykzLsViVO6LAUP5MSeGbEYNNVMnbrt9x+v
JJUEeKgDu+6B5dpffItKoZB0JaezPkvILFa9x8jvOOJckvB595yEunQtYQEgfn7R
8k8HWV+LLUNS60YMlOH1Zkd5d9VUWx+tJDfLRVpOoERIyNiwmcUVhAn21klJwGW4
5hpxbqCo8YLoRT5s1gLXCmeDBVrJpBA=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICCzCCAZGgAwIBAgISEdK7ujNu1LzmJGjFDYQdmOhDMAoGCCqGSM49BAMDMEYx
CzAJBgNVBAYTAkJFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMRwwGgYDVQQD
ExNHbG9iYWxTaWduIFJvb3QgRTQ2MB4XDTE5MDMyMDAwMDAwMFoXDTQ2MDMyMDAw
MDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYtc2Ex
HDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBFNDYwdjAQBgcqhkjOPQIBBgUrgQQA
IgNiAAScDrHPt+ieUnd1NPqlRqetMhkytAepJ8qUuwzSChDH2omwlwxwEwkBjtjq
R+q+soArzfwoDdusvKSGN+1wCAB16pMLey5SnCNoIwZD7JIvU4Tb+0cUB+hflGdd
yXqBPCCjQjBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1Ud
DgQWBBQxCpCPtsad0kRLgLWi5h+xEk8blTAKBggqhkjOPQQDAwNoADBlAjEA31SQ
7Zvvi5QCkxeCmb6zniz2C5GMn0oUsfZkvLtoURMMA/cVi4RguYv/Uo7njLwcAjA8
+RHUjE7AwWHCFUyqqx0LMV87HOIAl0Qx5v5zli/altP+CAezNIm8BZ/3Hobui3A=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFWjCCA0KgAwIBAgISEdK7udcjGJ5AXwqdLdDfJWfRMA0GCSqGSIb3DQEBDAUA
MEYxCzAJBgNVBAYTAkJFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMRwwGgYD
VQQDExNHbG9iYWxTaWduIFJvb3QgUjQ2MB4XDTE5MDMyMDAwMDAwMFoXDTQ2MDMy
MDAwMDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYt
c2ExHDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBSNDYwggIiMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCsrHQy6LNl5brtQyYdpokNRbopiLKkHWPd08EsCVeJ
OaFV6Wc0dwxu5FUdUiXSE2te4R2pt32JMl8Nnp8semNgQB+msLZ4j5lUlghYruQG
vGIFAha/r6gjA7aUD7xubMLL1aa7DOn2wQL7Id5m3RerdELv8HQvJfTqa1VbkNud
316HCkD7rRlr+/fKYIje2sGP1q7Vf9Q8g+7XFkyDRTNrJ9CG0Bwta/OrffGFqfUo
0q3v84RLHIf8E6M6cqJaESvWJ3En7YEtbWaBkoe0G1h6zD8K+kZPTXhc+CtI4wSE
y132tGqzZfxCnlEmIyDLPRT5ge1lFgBPGmSXZgjPjHvjK8Cd+RTyG/FWaha/LIWF
zXg4mutCagI0GIMXTpRW+LaCtfOW3T3zvn8gdz57GSNrLNRyc0NXfeD412lPFzYE
+cCQYDdF3uYM2HSNrpyibXRdQr4G9dlkbgIQrImwTDsHTUB+JMWKmIJ5jqSngiCN
I/onccnfxkF0oE32kRbcRoxfKWMxWXEM2G/CtjJ9++ZdU6Z+Ffy7dXxd7Pj2Fxzs
x2sZy/N78CsHpdlseVR2bJ0cpm4O6XkMqCNqo98bMDGfsVR7/mrLZqrcZdCinkqa

ByFrgY/bxFn63iLABJzjqls2k+g9vXqhnQt2sQvHnf3PmKgGwvgqo6GDoLclcqUC
4wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNV
HQ4EFgQUA1yrc4GHqMywptWU4jaWSf8FmSwwDQYJKoZIhvcNAQEMBQADggIBAHx4
7PYCLLtbfpIrXTncvtgdokIzTfnvpCo7RGkerNlFo048p9gkUbJUHJNOxO97k4Vg
JuoJSOD1u8fpaNK7ajFxzHmuEajwmf3lH7wvqMxX63bEIaZHU1VNaL8FpO7XJqti
2kM3S+LGteWygxk6x9PbTZ4IevPuzz5i+6zoYMzRx6Fcg0XERczzF2sUyQQCPtIk
pnnpHs6i58FZFZ8d4kuaPp92CC1r2LpXFNqD6v6MVenQTqnMdzGxRBF6XLE+0xRF
FRhiJBPSy03OXIPBNvIQtQ6IbbjhVp+J3pZmOUdkLG5NrmJ7v2B0GbhWrJKsFjLt
rWhV/pi60zTe9Mlhww6G9kuEYO4Ne7UyWHmRVSyBQ7N0H3qqJZ4d16GLuc1CLgSk
ZoNNiTW2bKg2SnkheCLQQrzRQDGQob4Ez8pn7fXwgNNgyYMqIgXQBztSvwyeqiv5
u+YfjyW6hY0XHgL+XVAEV8/+LbzvXMAaq7afJMbfc2hIkCwU9D9SGuTSyxTDYWnP
4vkYxboznxSjBF25cfe1lNj2M8FawTSLfJvdkzrnE6JwYZ+vj+vYxXX4M2bUdGc6
N3ec592kD3ZDZopD8p/7DEJ4Y9HiD2971KE9dJeFt0g5QdYg/NA6s/rob8SKunE3
vouXsXgxT7PntgMTzlSdriVZzH81Xwj3QEUxeCp6
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDxTCCAq2gAwIBAgIBADANBgkqhkiG9w0BAQsFADCBgzELMAkGA1UEBhMCVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxGjAYBgNVBAoT
EUdvRGFkZHkuY29tLCBJbmMuMTEwLwYDVQQDEyhHbyBEYWRkeSBSb290IENlcnRp
ZmljYXRlIEF1dGhvcml0eSAtIEcyMB4XDTA5MDkwMTAwMDAwMFoXDTM3MTIzMTIz
NTk1OVowgYMxCzAJBgNVBAYTAlVTMRAwDgYDVQQIEwdBcml6b25hMRMwEQYDVQQH
EwpTY290dHNkYWxlMRowGAYDVQQKExFHb0RhZGR5LmNvbSwgSW5jLjExMC8GA1UE
AxMoR28gRGFkZHkgUm9vdCBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkgLSBHMjCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL9xYgjx+lk09xvJGKP3gElY6SKD
E6bFIEMBO4Tx5oVJnyfq9oQbTqC023CYxzIBsQU+B07u9PpPL1kwIuerGVZr4oAH
/PMWdYA5UXvl+TW2dE6pjYIT5LY/qQOD+qK+ihVqf94Lw7YZFAXK6sOoBJQ7Rnwy
DfMAZiLIjWltNowRGLfTshxgtDj6AozO091GB94KPutdfMh8+7ArU6SSYmlRJQVh
GkSBjCypQ5Yj36w6gZoOKcUcqeldHraenjAKOc7xiID7S13MMuyFYkMlNAJWJwGR
tDtwKj9useiciAF9n9T521NtYJ2/LOdYq7hfRvzOxBsDPAnrSTFcaUaz4EcCAwEA
AaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYE
FDqahQcQZyi27/a9BUFuIMGU2g/eMA0GCSqGSIb3DQEBCwUAA4IBAQCZ21151fmX
WWcDYfF+OwYxdS2hII5PZYe096acvNjpL9DbWu7PdIxztDhC2gV7+AJ1uP2lsdeu
9tfeE8tTEH6KRtGX+rcuKxGrkLAngPnon1rpN5+r5N9ss4UXnT3ZJE95kTXWXwTr
gIOrmgIttRD02JDHBHNA7XIloKmf7J6raBKZV8aPEjoJpL1E/QYVN8Gb5DKj7Tjo
2GTzLH4U/ALqn83/B2gX2yKQOC16jdFU8WnjXzPKej17CuPKf1855eJ1usV2GDPO
LPAvTK33sefOT6jEm0pUBsV/fdUID+Ic/n4XuKxe9tQWskMJDE32p2u0mYRlynqI
4uJEvlz36hz1
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICVDCCAdugAwIBAgIQZ3SdjXfYO2rbIvT/WeK/zjAKBggqhkjOPQQDAzBsMQsw
CQYDVQQGEwJHUjE3MDUGA1UECgwuSGVsbGVuaWMgQWNhZGVtaWMgYW5kIFJlc2Vh
cmNoIEluc3RpdHV0aW9ucyBDQTEkMCIGA1UEAwwbSEFSSUNBIFRMUyBFQ0MgUm9v
dCBDQSAyMDIxMB4XDTIxMDIxOTExMDExMFoXDTQ1MDIxMzExMDEwOVowbDELMAkG
A1UEBhMCR1IxNzA1BgNVBAoMLkhlbGxlbmljIEFjYWRlbWljIGFuZCBSZXNlYXJj

aCBJbnN0aXR1dGlvbnMgQ0ExJDAiBgNVBAMMG0hBUklDQSBUTFMgRUNDIFJvb3Qg
Q0EgMjAyMTB2MBAGByqGSM49AgEGBSuBBAAiA2IABDgI/rGgltJ6rK9JOtDA4MM7
KKrxcm1lAEeIhPyaJmuqS7psBAqIXhfyVYf8MLA04jRYVxqEU+kw2anylnTDUR9Y
STHMmE5gEYd103KUkE+bECUqqHgtvpBBWJAVcqeht6NCMEAwDwYDVR0TAQH/BAUw
AwEB/zAdBgNVHQ4EFgQUyRtTgRL+BNUW0aq8mm+3oJUZbsowDgYDVR0PAQH/BAQD
AgGGMAoGCCqGSM49BAMDA2cAMGQCMBHervjcToiwqfAircJRQO9gcS3ujwLEXQNw
SaSS6sUUiHCm0w2wqsosQJz76YJumgIwK0eaB8bRwoF8yguWGEEbo/QwCZ61IygN
nxS2PFOiTAZpffpskcYqSUXm7LcT4Tps
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFpDCCA4ygAwIBAgIQOcqTHO9D88aOk8f0ZIk4fjANBgkqhkiG9w0BAQsFADBs
MQswCQYDVQQGEwJHUjE3MDUGA1UECgwuSGVsbGVuaWMgQWNhZGVtaWMgYW5kIFJl
c2VhcmNoIEluc3RpdHV0aW9ucyBDQTEkMCIGA1UEAwwbSEFSSUNBIFRMUyBSU0Eg
Um9vdCBDQSAyMDIxMB4XDTIxMDIxOTEwNTUzOFoXDTQ1MDIxMzEwNTUzN1owbDEL
MAkGA1UEBhMCR1IxNzA1BgNVBAoMLkhlbGxlbmljIEFjYWRlbWljIGFuZCBSZXNl
YXJjaCBJbnN0aXR1dGlvbnMgQ0ExJDAiBgNVBAMMG0hBUklDQSBUTFMgUlNBIFJv
b3QgQ0EgMjAyMTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAIvC569l
mwVnlskNJLnQDmT8zuIkGCyEf3dRywQRNrhe7Wlxp57kJQmXZ8FHws+RFjZiPTgE
4VGC/6zStGndLuwRo0Xua2s7TL+MjaQenRG56Tj5eg4MmOIjHdFOY9TnuEFE+2uv
a9of08WRiFukiZLRgeaMOVig1mlDqa2YUlhu2wr7a89o+uOkXjpFc5gH6l8Cct4M
pbOfrqkdtx2z/IpZ525yZa31MJQjB/OCFks1mJxTuy/K5FrZx40d/JiZ+yykgmvw
Kh+OC19xXFyuQnspiYHLA6OZyoieC0AJQTPb5lh6/a6ZcMBaD9YThnEvdmn8kN3b
LW7R8pv1GmuebxWMevBLKKAiOIAkbDakO/IwkfN4E8/BPzWr8R0RI7VDIp4BkrcY
AuUR0YLbFQDMYTfBKnya4dC6s1BG7oKsnTH4+yPiAwBIcKMJJnkVU2DzOFytOOqB
AGMUuTNe3QvboEUHGjMJ+E20pwKmafTCWQWIZYVWrkvL4N48fS0ayOn7H6NhStYq
E613TBoYm5EPWNgGVMWX+Ko/IIqmhaZ39qb8HOLubpQzKoNQhArlT4b4UEV4AIHr
W2jjJo3Me1xR9BQsQL4aYB16cmEdH2MtiKrOokWQCPxrvrNQKlr9qEgYRtaQQJKQ
CoReaDH46+0N0x3GfZkYVVYnZS6NRcUk7M7jAgMBAAGjQjBAMA8GA1UdEwEB/wQF
MAMBAf8wHQYDVR0OBBYEFApII6ZgpJIKM+qTW8VX6iVNvRLuMA4GA1UdDwEB/wQE
AwIBhjANBgkqhkiG9w0BAQsFAAOCAgEAPpBIqm5iFSVmewzVjIuJndftTgfvnNAU
X15QvWiWkKQUEapobQk1OUAJ2vQJLDSle1mESSmXdMgHHkdt8s4cUCbjnj1AUz/3
f5Z2EMVGpdAgS1D0NTsY9FVqQRtHBmg8uwkIYtlfVUKqrFOFrJVWNlar5AWMxaja
H6NpvVMPxP/cyuN+8kyIhkdGGvMA9YCRotxDQpSbIPDRzbLrLFPCU3hKTwSUQZqP
JzLB5UkZv/HywouoCjkxKLR9YjYsTewfM7Z+d21+UPCfDtcRj88YxeMn/ibvBZ3P
zzfF0HvaO7AWhAw6k9a+F9sPPg4ZeAnHqQJyIkv3N3a6dcSFA1pj1bF1BcK5vZSt
jBWZp5N99sXzqnTPBIWUmAD04vnKJGW/4GKvyMX6ssmeVkjaef2WdhW+o45WxLM0
/L5H9MG0qPzVMIho7suuyWPEdr6sOBjhXlzPrjoiUevRi7PzKzMHVIf6tLITe7pT
BGIBnfHAT+7hOtSLIBD6Alfm78ELt5BGnBkpjNxvoEppaZS3JGWg/6w/zgH7IS79
aPib8qXPMThcFarmlwDB31qlpzmq6YR/PFGoOtmUW4y/Twhx5duoXNTSpv4Ao8YW
xw/ogM4cKGR0GQjTQuPOAF1/sdwTsOEFy9EgqoZ0njnnkf3/W9b3raYvAwtt41dU
63ZTGI0RmLo=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICwzCCAkqgAwIBAgIBADAKBggqhkjOPQQDAjCBqjELMAkGA1UEBhMCR1IxDzAN

BgNVBAcTBkF0aGVuczFEMEIGA1UEChM7SGVsbGVuaWMgQWNhZGVtaWMgYW5kIFJl
c2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXRob3JpdHkxRDBCBgNVBAMTO0hl
bGxlbmljIEFjYWRlbWljIGFuZCBSZXNlYXJjaCBJbnN0aXR1dGlvbnMgRUNDIFJv
b3RDQSAyMDE1MB4XDTE1MDcwNzEwMzcxMloXDTQwMDYzMDEwMzcxMlowgaoxCzAJ
BgNVBAYTAkdSMQ8wDQYDVQQHEwZBdGhlbnMxRDBCBgNVBAoTO0hlbGxlbmljIEFj
YWRlbWljIGFuZCBSZXNlYXJjaCBJbnN0aXR1dGlvbnMgQ2VydC4gQXV0aG9yaXR5
MUQwQgYDVQQDEztIZWxsZW5pYyBBY2FkZW1pYyBhbmQgUmVzZWFyY2ggSW5zdGl0
dXRpb25zIEVDQyBSb290Q0EgMjAxNTB2MBAGByqGSM49AgEGBSuBBAAiA2IABJKg
QehLgoRc4vgxEZmGZE4JJS+dQS8KrjVPdJWyUWRrjWvmP3CV8AVER6ZyOFB2lQJa
jq4onvktTpnvLEhvTCUp6NFxW98dwXU3tNf6e3pCnGoKVlp8aQuqgAkkbH7BRqNC
MEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFLQi
C4KZJAEOnLvkDv2/+5cgk5kqMAoGCCqGSM49BAMCA2cAMGQCMGfOFmI4oqxiRaep
lSTAGiecMjvAwNW6qef4BENThe5SId6d9SWDPp5YSy/XZxMOIQIwBeF1Ad5o7Sof
TUwJCA3sS61kFyjndc5FZXIhF8siQQ6ME5g4mlRtm8rifOoCWCKR
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIGCzCCA/OgAwIBAgIBADANBgkqhkiG9w0BAQsFADCBpjELMAkGA1UEBhMCR1Ix
DzANBgNVBAcTBkF0aGVuczFEMEIGA1UEChM7SGVsbGVuaWMgQWNhZGVtaWMgYW5k
IFJlc2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXRob3JpdHkxQDA+BgNVBAMT
N0hlbGxlbmljIEFjYWRlbWljIGFuZCBSZXNlYXJjaCBJbnN0aXR1dGlvbnMgUm9v
dENBIDIwMTUwHhcNMTUwNzA3MTAxMTIxWhcNNDAwNjMwMTAxMTIxWjCBpjELMAkG
A1UEBhMCR1IxDzANBgNVBAcTBkF0aGVuczFEMEIGA1UEChM7SGVsbGVuaWMgQWNh
ZGVtaWMgYW5kIFJlc2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXRob3JpdHkx
QDA+BgNVBAMTN0hlbGxlbmljIEFjYWRlbWljIGFuZCBSZXNlYXJjaCBJbnN0aXR1
dGlvbnMgUm9vdENBIDIwMTUwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoIC
AQDC+Kk/G4n8PDwEXT2QNrCROnk8ZlrvbTkBSRq0t89/TSNTt5AA4xMqKKYx8ZEA
4yjsriFBzh/a/X0SWwGDD7mwX5nh8hKDgE0GPt+sr+ehiGsxr/CL0BgzuNtFajT0
AoAkKAoCFZVedioNmToUW/bLy1O8E00BiDeUJRtCvCLYjqOWXjrZMts+6PAQZe10
4S+nfK8nNLspfZu2zwnI5dMK/IhlZXQK3HMcXM1AsRzUtoSMTFDPaI6oWa7CJ06C
ojXdFPQf/7J31Ycvqm59JCfnxssm5uX+Zwdj2EUN3TpZZTlYepKZcj2chF6IIbjV
9Cz82XBST3i4vTwri5WY9bPRaM8gFH5MXF/ni+X1NYEZN9cRCLdmvtNKzoNXADrD
gfgXy5I2XdGj2HUb4Ysn6npIQf1FGQatJ5lOwXBH3bWfgVMS5bGMSF0xQxfjjMZ6
Y5ZLKTBOhE5iGV48zpeQpX8B653g+IuJ3SWYPZK2fu/Z8VFRfS0myGlZYeCsargq
NhEEelC9MoS+L9xy1dcdFkfkR2YgP/SWxa+OAXqlD3pk9Q0Yh9muiNX6hME6wGko
LfINaFGq46V3xqSQDqE3izEjR8EJCOtu93ib14L8hCCZSRm2Ekax+0VVFqmjZayc
Bw/qa9wfLgZy7IaIEuQt218FL+TwA9MmM+eAws1CoRc0CwIDAQABo0IwQDAPBgNV
HRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUcRVnyMjJvXVd
ctA4GGqd83EkVAswDQYJKoZIhvcNAQELBQADggIBAHW7bVRLqhBYRjTyYtcWNl0I
XtVsyIe9tC5G8jH4fOpCtZMWVdyhDBKg2mF+D1hYc2Ryx+hFjtyp8iY/xnmMsVMI
M4GwVhO+5lFc2JsKT0ucVlMC6U/2DWDqTUJV6HwbISHTGzrMd/K4kPFox/la/vot
9L/J9UUbzjgQKjeKeaO04wlshYaT/4mWJ3iBj2fjRnRUjtkNaeJK9E10A/+yd+2V
Z5fkscWrv2oj6NSU4kQoYsRL4vDY4ilrGnB+JGGTe08DMiUNRSQrlrRGar9KC/ea
j8GsGsVn82800vpzY4zvFrCopEYq+OsS7HK07/grfoxSwIuEVPkvPuNVqNxmsdnh
X9izjFk0WaSrT2y7HxjbdavYy5LNlDhhDgcGH0tGEPEVvo2FXDtKK4F5D7Rpn0lQ
l033DlZdwJVqwjbDG2jJ9SrcR5q+ss7FJej6A7na+RZukYT1HCjI/CbM1xyQVqdf
bzoEvM14iQuODy+jqk+iGxI9FghAD/FGTNeqewjBCvVtJ94Cj8rDtSvK6evIIVM4

pcw72Hc3MKJP2W/R8kCtQXoXxdZKNYm3QdV8hn9VTYNKpXMgwDqvkPGaJI7ZjnHK
e7iG2rKPmT4dEw0SEe7Uq/DpFXYC5ODfqiAeW2GFZECpkJcNrVPSWh2HagCXZWK0
vm9qp/UsQu0yrbYhnr68
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFajCCA1KgAwIBAgIQLd2szmKXlKFD6LDNdmpeYDANBgkqhkiG9w0BAQsFADBP
MQswCQYDVQQGEwJUVzEjMCEGA1UECgwaQ2h1bmdod2EgVGVsZWNvbSBDby4sIEx0
ZC4xGzAZBgNVBAMMEkhpUEtJIFJvb3QgQ0EgLSBHMTAeFw0xOTAyMjIwOTQ2MDRa
Fw0zNzEyMzExNTU5NTlaME8xCzAJBgNVBAYTAlRXMSMwIQYDVQQKDBpDaHVuZ2h3
YSBUZWxlY29tIENvLiwgTHRkLjEbMBkGA1UEAwwSSGlQS0kgUm9vdCBDQSAtIEcx
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA9B5/UnMyDHPkvRN0o9Qw
qNCuS9i233VHZvR85zkEHmpwINJaR3JnVfSl6J3VHiGh8Ge6zCFovkRTv4354twv
Vcg3Px+kwJyz5HdcoEb+d/oaoDjq7Zpy3iu9lFc6uux55199QmQ5eiY29yTw1S+6
lZgRZq2XNdZ1AYDgr/SEYYwNHl98h5ZeQa/rh+r4XfEuiAU+TCK72h8q3VJGZDnz
Qs7ZngyzsHeXZJzA9KMuH5UHsBffMNsAGJZMoYFL3QRtU6M9/Aes1MU3guvklQgZ
KILSQjqj2FPseYlgSGDIcpJQ3AOPgz+yQlda22rpEZfdhSi8MEyr48KxRURHH+CK
FgeW0iEPU8DtqX7UTuybCeyvQqww1r/REEXgphaypcXTT3OUM3ECoWqj1jOXTyFj
HluP2cFeRXF3D4FdXyGarYPM+l7WjSNfGz1BryB1ZlpK9p/7qxj3ccC2HTHsOyDr
y+K49a6SsvfhhEvyovKTmiKe0xRvNlS9H15ZFblzqMF8b3ti6RZsR1pl8w4Rm0bZ
/W3c1pzAtH2lsN0/Vm+h+fbkEkj9Bn8SV7apI09bA8PgcSojt/ewsTu8mL3WmKgM
a/aOEmem8rJY5AIJEzypuxC00jBF8ez3ABHfZfjcK0NVvxaXxA/VLGGEqnKG/uY6
fsI/fe78LxQ+5oXdUG+3Se0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAdBgNV
HQ4EFgQU8ncX+l6o/vY9cdVouslGDDjYr7AwDgYDVR0PAQH/BAQDAgGGMA0GCSqG
SIb3DQEBCwUAA4ICAQBQUfB13HAE4/+qddRxosuej6ip0691x1TPOhwEmSKsxBHi
7zNKpiMdDg1H2DfHb680f0+BazVP6XKlMeJ45/dOlBhbQH3PayFUhuaVevvGyuqc
SE5XCV0vrPSltJczWNWseanMX/mF+lLFjfiRFOs6DRfQUsJ748JzjkZ4Bjgs6Fza
ZsT0pPBWGTMpWmWSBUdGSquEwx4noR8RkpkndZMPvDY7l1ePJlsMu5wP1G4wB9Tc
XzZoZjmDlicmisjEOf6aIW/Vcobpf2Lll07QJNBAsNB1CI69aO4I1258EHBGG3zg
iLKecoaZAeO/n0kZtCW+VmWuF2PlHt/o/0elv+EmBYTksMCv5wiZqAxeJoBF1Pho
L5aPruJKHJwWDBNvOIf2u8g0X5IDUXlwpt/L9ZlNec1OvFefQ05rLisY+GpzjLrF
Ne85akEez3GoorKGB1s6yeHvP2UEgEcyRHCVTjFnanRbEEV16rCf0OY1/k6fi8wr
kkVbbiVghUbN0aqwdmaTd5a+g744tiROJgvM7XpWGuDpWsZkrUx6AEhEL7lAuxM+
vhV4nYWBSipX3tUZQ9rbyltHhoMLP7YNdnhzeSJesYAfz77RP1YQmCuVh6EfnWQU
YDksswBVLuT1sw5XxJFBAJw/6KXf6vb/yPCtbVKoF6ubYfwSUTXkJf2vqmqGOQ==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFzzCCA7egAwIBAgIUCBZfikyl7ADJk0DfxMauI7gcWqQwDQYJKoZIhvcNAQEL
BQAwbzELMAkGA1UEBhMCSEsxEjAQBgNVBAgTCUhvbmcgS29uZzESMBAGA1UEBxMJ
SG9uZyBLb25nMRYwFAYDVQQKEw1Ib25na29uZyBQb3N0MSAwHgYDVQQDExdIb25n
a29uZyBQb3N0IFJvb3QgQ0EgMzAeFw0xNzA2MDMwMjI5NDZaFw00MjA2MDMwMjI5
NDZaMG8xCzAJBgNVBAYTAkhLMRIwEAYDVQQIEwlIb25nIEtvbmcxEjAQBgNVBAcT
CUhvbmcgS29uZzEWMBQGA1UEChMNSG9uZ2tvbmcgUG9zdDEgMB4GA1UEAxMXSG9u
Z2tvbmcgUG9zdCBSb290IENBIDMwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIK
AoICAQCziNfqzg8gTr7m1gNt7ln8wlffKWihgw4+aMdoWJwcYEuJQwy51BWy7sFO

dem1p+/l6TWZ5Mwc50tfjTMwIDNT2aa71T4Tjukfh0mtUC1Qyhi+AViiE3CWu4mI
VoBc+L0sPOFMV4i707mV78vH9toxdCim5lSJ9UExyuUmGs2C4HDaOym71QP1mbpV
9WTRYA6ziUm4ii8F0oRFKHyPaFASePwLtVPLwpgchKOesL4jpNrcyCse2m5FHomY
2vkALgbpDDtw1VAliJnLzXNg99X/NWfFobxeq81KuEXryGgeDQ0URhLj0mRiikKY
vLTGCAj4/ahMZJx2Ab0vqWwzD9g/KLg8aQFChn5pwckGyuV6RmXpwtZQQS4/t+Tt
bNe/JgERohYpSms0BpDsE9K2+2p20jzt8NYt3eEV7KObLyzJPivkaTv/ciWxNoZb
x39ri1UbSsUgYT2uy1DhCDq+sI9jQVMwCFk8mB13umOResoQUGC/8Ne8lYePl8X+
l2oBlKN8W4UdKjk60FSh0Tlxnf0h+bV78OLgAo9uliQlLKAeLKjEiafv7ZkGL7YK
TE/bosw3Gq9HhS2KX8Q0NEwA/RiTZxPRN+ZItIsGxVd7GYYKecsAyVKvQv83j+Gj
Hno9UKtjBucVtT+2RTeUN7F+8kjDf8V1/peNRY8apxpyKBpADwIDAQABo2MwYTAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAfBgNVHSMEGDAWgBQXnc0e
i9Y5K3DTXNSguB+wAPzFYTAdBgNVHQ4EFgQUF53NHovWOStw01zUoLgfsAD8xWEw
DQYJKoZIhvcNAQELBQADggIBAFbVe27mIgHSQpsY1Q7XZiNc4/6gx5LS6ZStS6LG
7BJ8dNVI0lkUmcDrudHr9EgwW62nV3OZqdPlt9EuWSRY3GguLmLYauRwCy0gUCCk
MpXRAJi70/33MvJJrsZ64Ee+bs7Lo3I6LWldy8joRTnU+kLBEUx3XZL7av9YROXr
gZ6voJmtvqkBZss4HTzfQx/0TW60uhdG/H39h4F5ag0zD/ov+BS5gLNdTaqX4fnk
GMX41TiMJjz98iji7lpJiCzfeT2OnpA8vUFKOt1b9pq0zj8lMH8yfaIDlNDceqFS
3m6TjRgm/VWsvY+b0s+v54Ysyx8Jb6NvqYTUc79NoXQbTiNg8swOqn+knEwlqLJm
Ozj/2ZQw9nKEvmhVEA/GcywWaZMH/rFF7buiVWqw2rVKAiUnhde3t4ZEFolsgCs+
l6mc1X5VTMbeRRAc6uk7nwNT7u56AQIWeNTowr5GdogTPyK7SBIdUgC0An4hGh6c
JfTzPV4e0hz5sy229zdcxsshTrD3mUcYhcErulWuBurQB7Lcq9CClnXO0lD+mefP
L5/ndtFhKvshuzHQqp9HpLIiyhY6UFfEW0NnxWViA0kB60PZ2Pierc+xYw5F9KBa
LJstxabArahH9CdMOA0uG0k7UvToiIMrVCjU8jVStDKDYmlkDJGcn5fqdBb9HxEG
mpv0
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFazCCA1OgAwIBAgIRAIIQz7DSQONZRGPgu2OCiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh
cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMTUwNjA0MTEwNDM4
WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCcGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWFyY2ggR3JvdXAxFTATBgNVBAMTDElTUkcgUm9vdCBY
MTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK3oJHP0FDfzm54rVygc
h77ct984kIxuPOZXoHj3dcKi/vVqbvYATyjb3miGbESTtrFj/RQSa78f0uoxmyF+
0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpmZmyPfjxwv60pIgbz5MDmgK7iS4+3mX6U
A5/TR5d8mUgjU+g4rk8Kb4Mu0UlXjIB0ttov0DiNewNwIRt18jA8+o+u3dpjq+sW
T8KOEUt+zwvo/7V3LvSye0rgTBIlDHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyH
B5T0Y3HsLuJvW5iB4YlcNHlsdu87kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UC
B5iPNgiV5+I3lg02dZ77DnKxHZu8A/lJBdiB3QW0KtZB6awBdpUKD9jf1b0SHzUv
KBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKT5ZPt0m9STJEadao0xAH0ahmbWn
OlFuhjuefXKnEgV4We0+UXgVCwOPjdAvBbI+e0ocS3MFEvzG6uBQE3xDk3SzynTn
jh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9gwQbooMDQaHWBfEbwrbw
qHyGO0aoSCqI3Haadr8faqU9GY/rOPNk3sgrDQoo//fb4hVC1CLQJ13hef4Y53CI
rU7m2Ys6xt0nUW7/vGT1M0NPAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBBR5tFnme7bl5AFzgAiIyBpY9umbbjANBgkq
hkiG9w0BAQsFAAOCAgEAVR9YqbyyqFDQDLHYGmkgJykIrGF1XIpu+ILlaS/V9lZL
ubhzEFnTIZd+50xx+7LSYK05qAvqFyFWhfFQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ

3BebYhtF8GaV0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiBOv/2X/qkSsisRcOj/KK
NFtY2PwByVS5uCbMiogziUwthDyC3+6WVwW6LLv3xLfHTjuCvjHIInNzktHCgKQ5
ORAzI4JMPJ+GslWYHb4phowim57iaztXOoJwTdwJx4nLCgdNbOhdjsnvzqvHu7Ur
TkXWStAmzOVyyghqpZXjFaH3pO3JLF+l+/+sKAIuvtd7u+Nxe5AW0wdeRlN8NwdC
jNPElpzVmbUq4JUagEiuTDkHzsxHpFKVK7q4+63SM1N95R1NbdWhscdCb+ZAJzVc
oyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMljq4Ui0/1lvh+wjChP4kqKOJ2qxq
4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d11TPA
mRGunUHBcnWEvgJBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d
emyPxgcYxn/eR44/KJ4EBs+lVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iItreGCc=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICGzCCAaGgAwIBAgIQQdKd0XLq7qeAwSxs6S+HUjAKBggqhkjOPQQDAzBPMQsw
CQYDVQQGEwJVUzEpMCcGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWFyY2gg
R3JvdXAxFTATBgNVBAMTDElTUkcgUm9vdCBYMjAeFw0yMDA5MDQwMDAwMDBaFw00
MDA5MTcxNjAwMDBaME8xCzAJBgNVBAYTAlVTMSkwJwYDVQQKEyBJbnRlcm5ldCBT
ZWN1cml0eSBSZXNlYXJjaCBHcm91cDEVMBMGA1UEAxMMSVNSRyBSb290IFgyMHYw
EAYHKoZIzj0CAQYFK4EEACIDYgAEzZvVn4CDCuwJSvMWSj5cz3es3mcFDR0HttwW
+1qLFNvicWDEukWVEYmO6gbf9yoWHKS5xcUy4APgHoIYOIvXRdgKam7mAHf7AlF9
ItgKbppbd9/w+kHsOdx1ymgHDB/qo0IwQDAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0T
AQH/BAUwAwEB/zAdBgNVHQ4EFgQUfEKWrt5LSDv6kviejM9ti6lyN5UwCgYIKoZI
zj0EAwMDaAAwZQIwe3lORlCEwkSHRhtFcP9Ymd70/aTSVaYgLXTWNLxBo1BfASdW
tL4ndQavEi51mI38AjEAi/V3bNTIZargCyzuFJ0nN6T5U6VR5CmD1/iQMVtCnwr1
/q4AaOeMSQ+2b1tbFfLn
-----END CERTIFICATE-----

-----BEGIN
CERTIFICATE-----
MIIFYDCCA0igAwIBAgIQCgFCgAAAAUUjyES1AAAAAjANBgkqhkiG9w0BAQsFADBK
MQswCQYDVQQGEwJVUzESMBAGA1UEChMJSWRlblRydXN0MScwJQYDVQQDEx5JZGVu
VHJ1c3QgQ29tbWVyY2lhbCBSb290IENBIDEwHhcNMTQwMTE2MTgxMjIzWhcNMzQw
MTE2MTgxMjIzWjBKMQswCQYDVQQGEwJVUzESMBAGA1UEChMJSWRlblRydXN0MScw
JQYDVQQDEx5JZGVuVHJ1c3QgQ29tbWVyY2lhbCBSb290IENBIDEwggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCnUBneP5k91DNG8W9RYYKyqU+PZ4ldhNlT
3Qwo2dfw/66VQ3KZ+bVdfIrBQuExUHTRgQ18zZshq0PirK1ehm7zCYofWjK9ouuU
+ehcCuz/mNKvcbO0U59Oh++SvL3sTzIwiEsXXlfEU8L2ApeN2WIrvyQfYo3fw7gp
S0l4PJNgiCL8mdo2yMKi1CxUAGc1bnO/AljwpN3lsKImesrgNqUZFvX9t++uP0D1
bVoE/c40yiTcdCMbXTMTEl3EASX2MN0CXZ/g1Ue9tOsbobtJSdifWwLziuQkkORi
T0/Br4sOdBeo0XKIanoBScy0RnnGF7HamB4HWfp1IYVl3ZBWzvurpWCdxJ35UrCL
vYf5jysjCiN2O/cz4ckA82n5S6LgTrx+kzmEB/dEcH7+B1rlsazRGMzyNeVJSQjK
Vsk9+w8YfYs7wRPCTY/JTw436R+hDmrfYi7LNQZReSzIJTj0+kuniVyc0uMNOYZK
dHzVWYfCP04MXFL0PfdSgvHqo6z9STQaKPNBiDoT7uje/5kdX7rL6B7yuVBgwDHT
c+XvvqDtMwt0viAgxGds8AgDelWAf0ZOlqf0Hj7h9tgJ4TNkK2PXMl6f+cB7D3hv
l7yTmvmcEpB4eoCHFddydJxVdHixuuFucAS6T6C6aMN7/zHwcz09lCqxC0EOoP5N
iGVreTO01wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB
/zAdBgNVHQ4EFgQU7UQZwNPwBovupHu+QucmVMiONnYwDQYJKoZIhvcNAQELBQAD
ggIBAA2ukDL2pkt8RHYZYR4nKM1eVO8lvOMIkPkp165oCOGUAFjvLi5+U1KMtlwH

6oi6mYtQlNeCgN9hCQCTrQ0U5s7B8jeUeLBfnLOic7iPBZM4zY0+sLj7wM+x8uwt
LRvM7Kqas6pgghstO8OEPVeKlh6cdbjTMM1gCIOQ045U8U1mwF10A0Cj7oV+wh93
nAbowacYXVKV7cndJZ5t+qntozo00Fl72u1Q8zW/7esUTTHHYPTa8Yec4kjixsU3
+wYQ+nVZZjFHKdp2mhzpgq7vmrlR94gjmmmVYjzlVYA211QC//G5Xc7UI2/YRYRK
W2XviQzdFKcgyxilJbQN+QHwotL0AMh0jqEqSI5l2xPE4iUXfeu+h1sXIFRRk0pT
AwvsXcoz7WL9RccvW9xYoIA55vrX/hMUpu09lEpCdNTDd1lzzY9GvlU47/rokTLq
l1gEIt44w8y8bckzOmoKaT+gyOpyj4xjhiO9bTyWnpXgSUyqorkqG5w2gXjtw+hG
4iZZRHUe2XWJUc0QhJ1hYMtd+ZciTY6Y5uN/9lu7rs3KSoFrXgvzUeF0K+l+J6fZ
mUlO+KWA2yUPHGNiiskzZ2s8EIPGrd6ozRaOjfAHN3Gf8qv8QfXBi+wAN10J5U6A
7/qxXDgGpRtK4dw4LTzcqx+QGtVKnO7RcGzM7vRX+Bi6hG6H
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFZjCCA06gAwIBAgIQCgFCgAAAAUUjz0Z8AAAAAjANBgkqhkiG9w0BAQsFADBN
MQswCQYDVQQGEwJVUzESMBAGA1UEChMJSWRlblRydXN0MSowKAYDVQQDEyFJZGVu
VHJ1c3QgUHVibGljIFNlY3RvciBSb290IENBIDEwHhcNMTQwMTE2MTc1MzMyWhcN
MzQwMTE2MTc1MzMyWjBNMQswCQYDVQQGEwJVUzESMBAGA1UEChMJSWRlblRydXN0
MSowKAYDVQQDEyFJZGVuVHJ1c3QgUHVibGljIFNlY3RvciBSb290IENBIDEwggIi
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC2IpT8pEiv6EdrCvsnduTyP4o7
ekosMSqMjbCpwzFrqHd2hCa2rIFCDQjrVVi7evi8ZX3yoG2LqEfpYnYeEe4IFNGy
RBb06tD6Hi9e28tzQa68ALBKK0CyrOE7S8ItneShm+waOh7wCLPQ5CQ1B5+ctMlS
bdsHyo+1W/CD80/HLaXIrcuVIKQxKFdYWuSNG5qrng0M8gozOSI5Cpcu81N3uURF
/YTLNiCBWS2ab21ISGHKTN9T0a9SvESfqy9rg3LvdYDaBjMbXcjaY8ZNzaxmMc3R
3j6HEDbhuaR672BQssvKplbgN6+rNBM5Jeg5ZuSYeqoSmJxZZoY+rfGwyj4GD3vw
EUs3oERte8uojHH01bWRNszwFcYr3lEXsZdMUD2xlVl8BX0tIdUAvwFnol57plzy
9yLxkA2T26pEUWbMfXYD62qoKjgZl3YNa4ph+bz27nb9cCvdKTz4Ch5bQhyLVi9V
GxyhLrXHFub4qjySjmm2AcG1hp2JDws4lFTo6tyePSW8Uybt1as5qsVATFSrsrTZ
2fjXctscvG29ZV/viDUqZi/u9rNl8DONfJhBaUYPQxxp+pu10GFqzcpL2UyQRqsV
WaFHVCkugyhfHMKiq3IXAAaOReyL4jM9f9oZRORicsPfIsbyVtTdX5Vy7W1f90gD
W/3FKqD2cyOEEBsB5wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/
BAUwAwEB/zAdBgNVHQ4EFgQU43HgntinQtnbcZFrlJPrw6PRFKMwDQYJKoZIhvcN
AQELBQADggIBAEf63QqwEZE4rU1d9+UOl1QZgkiHVIyqZJnYWv6IAcVYpZmxI1Qj
t2odIFflAWJBF9MJ23XLblSQdf4an4EKwt3X9wnQW3IV5B4Jaj0z8yGa5hV+rVHV
DRDtfULAj+7AmgjVQdZcDiFpboBhDhXAuM/FSRJSzL46zNQuOAXeNf0fb7iAaJg9
TaDKQGXSc3z1i9kKlT/YPyNtGtEqJBnZhbMX73huqVjRI9PHE+1yJX9dsXNw0H8G
lwmEKYBhHfpe/3OsoOOJuBxxFcbeMX8S3OFtm6/n6J91eEyrRjuazr8FGF1NFTwW
mhlQBJqymm9li1JfPFgEKCXAZmExfrngdbkaqIHWchezxQMxNRF4eKLg6TCMf4Df
WN88uieW4oA0beOY02QnrEh+KHdcxiVhJfiFDGX6xDIvpZgF5PgLZxYWxoK4Mhn5
+bl53B/N66+rDt0b20XkeucC4pVd/GnwU2lhlXV5C15V5jgclKlZM57IcXR5f1GJ
tshquDDIajjDbp7hNxbqBWJMWxJH7ae0s1hWx0nzfxJoCTFx8G34Tkf71oXuxVhA
GaQdp/lLQzfcaFpPz+vCZHTetBXZ9FRUGi8c15dxVJCO2SCdUyt/q4/i6jC8UDfv
8Ue1fXwsBOxonbRJRBD0ckscZOf85muQ3Wl9af0AVqW3rLatt8o+Ae+c
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF8TCCA9mgAwIBAgIQALC3WhZIX7/hy/WL1xnmfTANBgkqhkiG9w0BAQsFADA4

MQswCQYDVQQGEwJFUzEUMBIGA1UECgwLSVpFTlBFIFMuQS4xEzARBgNVBAMMCkl6
ZW5wZS5jb20wHhcNMDcxMjEzMTMwODI4WhcNMzcxMjEzMDgyNzI1WjA4MQswCQYD
VQQGEwJFUzEUMBIGA1UECgwLSVpFTlBFIFMuQS4xEzARBgNVBAMMCkl6ZW5wZS5j
b20wggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDJ03rKDx6sp4boFmVq
scIbRTJxldn+EFvMr+eleQGPicPK8lVx93e+d5TzcqQsRNiekpsUOqHnJJAKClaO
xdgmlOHZSOEtPtoKct2jmRXagaKH9HtuJneJWK3W6wyyQXpzbm3benhB6QiIEn6H
LmYRY2xU+zydcsC8Lv/Ct90NduM61/e0aL6i9eOBbsFGb12N4E3GVFWJGjMxCrFX
uaOKmMPsOzTFlUFpfnXCPCDFYbpRR6AgkJOhkEvzTnyFRVSa0QUmQbC1TR0zvsQD
yCV8wXDbO/QJLVQnSKwv4cSsPsjLkkxTOTcj7NMB+eAJRE1NZMDhDVqHIrytG6P+
JrUV86f8hBnp7KGItERphIPzidF0BqnMC9bC3ieFUCbKF7jJeodWLBoBHmy+E60Q
rLUk9TiRodZL2vG70t5HtfG8gfZZa88ZU+mNFctKy6lvROUbQc/hhqfK0GqfvEyN
BjNaooXlkDWgYlwWTvDjovoDGrQscbNYLN57C9saD+veIR8GdwYDsMnvmfzAuU8L
hij+0rnq49qlw0dpEuDb8PYZi+17cNcC1u2HGCgsBCRMd+RIihrGO5rUD8r6ddIB
QFqNeb+Lz0vPqhbBleStTIo+F5HUsWLlguWABKQDfo2/2n+iD5dPDNMN+9fR5XJ+
HMh3/1uaD7euBUbl8agW7EekFwIDAQABo4H2MIHzMIGwBgNVHREEgagwgaWBD2lu
Zm9AaXplbnBlLmNvbaSBkTCBjjFHMEUGA1UECgw+SVpFTlBFIFMuQS4gLSBDSUYg
QTAxMzM3MjYwLVJNZXJjLlZpdG9yaWEtR2FzdGVpeiBUMTA1NSBGNjIgUzgxQzBB
BgNVBAkMOkF2ZGEgZGVsIE1lZGl0ZXJyYW5lbyBFdG9yYmlkZWEgMTQgLSAwMTAx
MCBWaXRvcmlhLUdhc3RlaXowDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AQYwHQYDVR0OBBYEFB0cZQ6o8iV7tJHP5LGx5r1VdGwFMA0GCSqGSIb3DQEBCwUA
A4ICAQB4pgwWSp9MiDrAyw6lFn2fuUhfGI8NYjb2zRlrrKvV9pF9rnHzP7MOeIWb
laQnIUdCSnxIOvVFfLMMjlF4rJUT3sb9fbgakEyrkgPH7UIBzg/YsfqikuFgba56
awmqxinuaElnMIAkejEWOVt+8Rwu3WwJrfIxwYJOubv5vr8qhT/AQKM6WfxZSzwo
JNu0FXWuDYi6LnPAvViH5ULy617uHjAimcs30cQhbIHsvm0m5hzkQiCeR7Csg1lw
LDXWrzY0tM07+DKo7+N4ifuNRSzanLh+QBxh5z6ikixL8s36mLYp//Pye6kfLqCT
VyvehQP5aTfLnnhqBbTFMXiJ7HqnheG5ezzevh55hM6fcA5ZwjUukCox2eRFekGk
LhObNA5me0mrZJfQRsN5nXJQY6aYWwa9SG3YOYNw6DXwBdGqvOPbyALqfP2C2sJb
UjWumDqtujWTI6cfSN01RpiyEGjkpTHCClguGYEQyVB1/OpaFs4R1+7vUIgtYf8/
QnMFlEPVjjxOAToZpR9GTnfQXeWBIiGH/pR9hNiTrdZoQ0iy2+tzJOeRf1SktoA+
naM8THLCV8Sg1Mw4J87VBp6iSNnpn86CcDaTmjvfliHjWbcM2pE38P1ZWrOZyGls
QyYBNWNgVYkDOnXYukrZVP/u3oDYLdE41V4tC5h9Pmzb/CaIxw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIECjCCAvKgAwIBAgIJAMJ+QwRORz8ZMA0GCSqGSIb3DQEBCwUAMIGCMQswCQYD
VQQGEwJIVTERMA8GA1UEBwwIQnVkYXBlc3QxFjAUBgNVBAoMDU1pY3Jvc2VjIEx0
ZC4xJzAlBgNVBAMMHk1pY3Jvc2VjIGUtU3ppZ25vIFJvb3QgQ0EgMjAwOTEfMB0G
CSqGSIb3DQEJARYQaW5mb0BlLXN6aWduby5odTAeFw0wOTA2MTYxMTMwMThaFw0y
OTEyMzAxMTMwMThaMIGCMQswCQYDVQQGEwJIVTERMA8GA1UEBwwIQnVkYXBlc3Qx
FjAUBgNVBAoMDU1pY3Jvc2VjIEx0ZC4xJzAlBgNVBAMMHk1pY3Jvc2VjIGUtU3pp
Z25vIFJvb3QgQ0EgMjAwOTEfMB0GCSqGSIb3DQEJARYQaW5mb0BlLXN6aWduby5o
dTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOn4j/NjrdqG2KfgQvvP
kd6mJviZpWNwrZuuyjNAfW2WbqEORO7hE52UQlKavXWFdCyoDh2Tthi3jCyoz/tc
cbna7P7ofo/kLx2yqHWH2Leh5TvPmUpG0IMZfcChEhyVbUr02MelTTMuhTlAdX4U
fIASmFDHQWe4oIBhVKZsTh/gnQ4H6cm6M+f+wFUoLAKApxn1ntxVUwOXewdI/5n7
N4okxFnMUBBjjqqpGrCEGob5X7uxUG6k0QrM1XF+H6cbfPVTbiJfyyvm1HxdrtbC
xkzlBQHZ7Vf8wSN5/PrIJIOV87VqUQHQd9bpEqH5GoP7ghu5sJf0dgYzQ0mg/wu1

+rUCAwEAAaOBgDB+MA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0G
A1UdDgQWBBTLD8bfQkPMPcu1SCOhGnqmKrs0aDAfBgNVHSMEGDAWgBTLD8bfQkPM
Pcu1SCOhGnqmKrs0aDAbBgNVHREEFDASgRBpbmZvQGUtc3ppZ25vLmh1MA0GCSqG
SIb3DQEBCwUAA4IBAQDJ0Q5eLtXMs3w+y/w9/w0olZMEyL/azXm4Q5DwpL7v8u8h
mLzU1F0G9u5C7DBsoKqpyvGvivo/C3NqPuouQH4frlRheesuCDfXI/OMn74dseGk
ddug4lQUsbocKaQY9hK6ohQU4zE1yED/t+AFdlfBHFny+L/k7SViXITwfn4fs775
tyERzAMBVnCnEJIeGzSBHq2cGsMEPO0CYdYeBvNfOofyK/FFh+U9rNHHV4S9a67c
2Pm2G2JwCz02yULyMtd6YebS2z3PyKnJm9zbWETXbzivf3jTo60adbocwTZ8jx5t
HMN1Rq41Bab2XD0h7lbwyYIiLXpUq3DDfSJlgnCW
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICWTCCAd+gAwIBAgIQZvI9r4fei7FK6gxXMQHC7DAKBggqhkjOPQQDAzBlMQsw
CQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9uMTYwNAYD
VQQDEy1NaWNyb3NvZnQgRUNDIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDIw
MTcwHhcNMTkxMjE4MjMwNjQ1WhcNNDIwNzE4MjMxNjA0WjBlMQswCQYDVQQGEwJV
UzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9uMTYwNAYDVQQDEy1NaWNy
b3NvZnQgRUNDIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDIwMTcwdjAQBgcq
hkjOPQIBBgUrgQQAIgNiAATUvD0CQnVBEyPNgASGAlEvaqiBYgtlzPbKnR5vSmZR
ogPZnZH6thaxjG7efM3beaYvzrvOcS/lpaso7GMEZpn4+vKTEAXhgShC48Zo9OYb
hGBKia/teQ87zvH2RPUBeMCjVDBSMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8E
BTADAQH/MB0GA1UdDgQWBBTIy5lycFIM+Oa+sgRXKSrPQhDtNTAQBgkrBgEEAYI3
FQEEAwIBADAKBggqhkjOPQQDAwNoADBlAjBY8k3qDPlfXu5gKcs68tvWMoQZP3zV
L8KxzJOuULsJMsbG7X7JNpQS5GiFBqIb0C8CMQCZ6Ra0DvpWSNSkMBaReNtUjGUB
iudQZsIxtzm6uBoiB078a1QWIP8rtedMDE2mT3M=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFqDCCA5CgAwIBAgIQHtOXCV/YtLNHcB6qvn9FszANBgkqhkiG9w0BAQwFADBl
MQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9uMTYw
NAYDVQQDEy1NaWNyb3NvZnQgUlNBIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5
IDIwMTcwHhcNMTkxMjE4MjI1MTIyWhcNNDIwNzE4MjMwMDIzWjBlMQswCQYDVQQG
EwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9uMTYwNAYDVQQDEy1N
aWNyb3NvZnQgUlNBIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDIwMTcwggIi
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDKW76UM4wplZEWCpW9R2LBifOZ
Nt9GkMml7Xhqb0eRaPgnZ1AzHaGm++DlQ6OEAlcBXZxIQIJTELy/xztokLaCLeX0
ZdDMbRnMlfl7rEqUrQ7eS0MdhweSE5CAg2Q1OQT85elss7YfUJQ4ZVBcF0a5toW1
HLUX6NZFndiyJrDKxHBKrmCk3bPZ7Pw71VdyvD/IybLeS2v4I2wDwAW9lcfNcztm
gGTjGqwu+UcF8ga2m3P1eDNbx6H7JyqhtJqRjJHTOoI+dkC0zVJhUXAoP8XFWvLJ
jEm7FFtNyP9nTUwSlq31/niol4fX/V4ggNyhSyL71Imtus5Hl0dVe49FyGcohJUc
aDDv70ngNXtk55iwlNpNhTs+VcQor1fznhPbRiefHqJeRIOkpcrVE7NLP8TjwuaG
YaRSMLl6IE9vDzhTyzMMEyuP1pq9KsgtsRx9S1HKR9FIJ3Jdh+vVReZIZZ2vUpC6
W6IYZVcSn2i51BVrlMRpIpj0M+Dt+VGOQVDJNE92kKz8OMHY4Xu54+OU4UZpyw4K
UGsTuqwPN1q3ErWQgR5WrlcihtnJ0tHXUeOrO8ZV/R4O03QK0dqq6mm4lyiPSMQH
+FJDOvTKVTUssKZqwJz58oHhEmrARdlns87/I6KJClTUFLkqqNfs+avNJVgyeY+Q
W5g5xAgGwax/Dj0ApQIDAQABo1QwUjAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/

BAUwAwEB/zAdBgNVHQ4EFgQUCctZf4aycI8awznjwNnpv7tNsiMwEAYJKwYBBAGC
NxUBBAMCAQAwDQYJKoZIhvcNAQEMBQADggIBAKyvPl3CEZaJjqPnktaXFbgToqZC
LgLNFgVZJ8og6Lq46BrsTaiXVq5lQ7GPAJtSzVXNUzltYkyLDVt8LkS/gxCP81OC
gMNPOsduET/m4xaRhPtthH80dK2Jp86519efhGSSvpWhrQlTM93uCupKUY5vVau6
tZRGrox/2KJQJWVggEbbMwSubLWYdFQl3JPk+ONVFT24bcMKpBLBaYVu32TxU5nh
SnUgnZUP5NbcA/FZGOhHibJXWpS2qdgXKxdJ5XbLwVaZOjex/2kskZGT4d9Mozd2
TaGf+G0eHdP67Pv0RR0Tbc/3WeUiJ3IrhvNXuzDtJE3cfVa7o7P4NHmJweDyAmH3
pvwPuxwXC65B2Xy9J6P9LjrRk5Sxcx0ki69bIImtt2dmefU6xqaWM/5TkshGsRGR
xpl/j8nWZjEgQRCHLQzWwa80mMpkg/sTV9HB8Dx6jKXB/ZUhoHHBk2dxEuqPiApp
GWSZI1b7rCoucL5mxAyE7+WL85MB+GqQk2dLsmijtWKP6T+MejteD+eMuMZ87zf9
dOLITzNy4ZQ5bb0Sr74MTnB8G2+NszKTc0QWbej09+CVgI+WXTik9KveCjCHk9hN
AHFiRSdLOkKEW39lt2c0Ui2cFmuqqNh7o0JMcccMyj6D5KbvtwEwXlGjefVwaaZB
RA+GsCyRxj3qrg+E
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFojCCA4qgAwIBAgIUAZQwHqIL3fXFMyqxQ0Rx+NZQTQ0wDQYJKoZIhvcNAQEM
BQAwaTELMAkGA1UEBhMCS1IxJjAkBgNVBAoMHU5BVkVSIEJVU0lORVNTIFBMQVRG
T1JNIENvcnAuMTIwMAYDVQQDDClOQVZFUiBHbG9iYWwgUm9vdCBDZXJ0aWZpY2F0
aW9uIEF1dGhvcml0eTAeFw0xNzA4MTgwODU4NDJaFw0zNzA4MTgyMzU5NTlaMGkx
CzAJBgNVBAYTAktSMSYwJAYDVQQKDB1OQVZFUiBCVVNJTkVTUyBQTEFURk9STSBD
b3JwLjEyMDAGA1UEAwwpTkFWRVIgR2xvYmFsIFJvb3QgQ2VydGlmaWNhdGlvbiBB
dXRob3JpdHkwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC21PGTXLVA
iQqrDZBbUGOukJR0F0Vy1ntlWilLp1agS7gvQnXp2XskWjFlqxcX0TM62RHcQDaH
38dq6SZeWYp34+hInDEW+j6RscrJo+KfziFTowI2MMtSAuXaMl3Dxeb57hHHi8lE
HoSTGEq0n+USZGnQJoViAbbJAh2+g1G7XNr4rRVqmfeSVPc0W+m/6imBEtRTkZaz
kVrd/pBzKPswRrXKCAfHcXLJZtM0l/aM9BhK4dA9WkW2aacp+yPOiNgSnABIqKYP
szuSjXEOdMWLyEz59JuOuDxp7W87UC9Y7cSw0BwbagzivESq2M0UXZR4Yb8Obtoq
vC8MC3GmsxY/nOb5zJ9TNeIDoKAYv7vxvvTWjIcNQvcGufFt7QSUqP620wbGQGHf
nZ3zVHbOUzoBppJB7ASjjw2i1QnK1sua8e9DXcCrpUHPXFNwcMmIpi3Ua2FzUCaG
YQ5fG8Ir4ozVu53BA0K6lNpfqbDKzE0K70dpAy8i+/Eozr9dUGWokG2zdLAIx6yo
0es+nPxdGoMuK8u180SdOqcXYZaicdNwlhVNt0xz7hlcxVs+Qf6sdWA7G2POAN3a
CJBitOUt7kinaxeZVL6HSuOpXgRM6xBtVNbv8ejyYhbLgGvtPe31HzClrkvJE+2K
AQHJuFFYwGY6sWZLxNUxAmLpdIQM201GLQIDAQABo0IwQDAdBgNVHQ4EFgQU0p+I
36HNLL3s9TsBAZMzJ7LrYEswDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMB
Af8wDQYJKoZIhvcNAQEMBQADggIBADLKgLOdPVQG3dLSLvCkASELZ0jKbY7gyKoN
qo0hV4/GPnrK21HUUrPUloSlWGB/5QuOH/XcChWB5Tu2tyIvCZwTFrFsDDUIbatj
cu3cvuzHV+YwIHHW1xDBE1UBjCpD5EHxzzp6U5LOogMFDTjfArsQLtk70pt6wKGm
+LUx5vR1yblTmXVHIloUFcd4G7ad6Qz4G3bxhYTeodoS76TiEJd6eN4MUZeoIUCL
hr0N8F5OSza7OyAfikJW4Qsav3vQIkMsRIz75Sq0bBwcupTgE34h5prCy8VCZLQe
lHsIJchxzIdFV4XTnyliIoNRlwAYl3dqmJLJfGBs32x9SuRwTMKeuB330DTHD8z7
p/8Dvq1wkNoL3chtl1+afwkyQf3NosxabUzyqkn+Zvjp2DXrDige7kgvOtB5CTh8
piKCk5XQA76+AqAF3SAi428diDRgxuYKuQl1C/AH6GmWNcf7I4GOODm4RStDeKLR
LBT/DShycpWbXgnbiUSYqqFJu3FS8r/2/yehNq+4tneI3TqkbZs0kNwUXTC/t+sX
5Ie3cdCh13cV1ELX8vMxmV2b3RZtP+oGI/hGoiLtk/bdmuYqh7GYVPEi92tF4+KO
dh2ajcQGjTa3FPOdVGm3jjzVpG2Tgbet9r1ke8LJaDmgkpzNNIaRkPpkUZ3+/uul
9XXeifdy

-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIEFTCCAv2gAwIBAgIGSUEs5AAQMA0GCSqGSIb3DQEBCwUAMIGnMQswCQYDVQQG
EwJIVTERMA8GA1UEBwwIQnVkYXBlc3QxFTATBgNVBAoMDE5ldExvY2sgS2Z0LjE3
MDUGA1UECwwuVGFuw7pzw610dsOhbnlraWFkw7NrIChDZXJ0aWZpY2F0aW9uIFNl
cnZpY2VzKTE1MDMGA1UEAwwsTmV0TG9jayBBcmFueSAoQ2xhc3MgR29sZCkgRsWR
dGFuw7pzw610dsOhbnkwHhcNMDgxMjExMTUwODIxWhcNMjgxMjA2MTUwODIxWjCB
pzELMAkGA1UEBhMCSFUxETAPBgNVBAcMCEJ1ZGFwZXN0MRUwEwYDVQQKDAxOZXRM
b2NrIEtmdC4xNzA1BgNVBAsMLlRhbsO6c8OtdHbDoW55a2lhZMOzayAoQ2VydGlm
aWNhdGlvbiBTZXJ2aWNlcykxNTAzBgNVBAMMLE5ldExvY2sgQXJhbnkgKENsYXNz
IEdvbGQpIEbFkXRhbsO6c8OtdHbDoW55MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAxCRec75LbRTDofTjl5Bu0jBFHjzuZ9lk4BqKf8owyoPjIMHj9DrT
lF8afFttvzBPhCf2nx9JvMaZCpDyD/V/Q4Q3Y1GLeqVw/HpYzY6b7cNGbIRwXdrz
AZAj/E4wqX7hJ2Pn7WQ8oLjJM2P+FpD/sLj916jAwJRDC7bVWaaeVtAkH3B5r9s5
VA1lddkVQZQBr17s9o3x/61k/iCa11zr/qYfCGSji3ZVrR47KGAuhyXoqq8fxmRG
ILdwfzzeSNuWU7c5d+Qa4scWhHaXWy+7GRWF+GmF9ZmnqfI0p6m2pgP8b4Y9VHx2
BJtr+UBdADTHLpl1neWIA6pN+APSQnbAGwIDAKiLo0UwQzASBgNVHRMBAf8ECDAG
AQH/AgEEMA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUzPpnk/C2uNClwB7zU/2M
U9+D15YwDQYJKoZIhvcNAQELBQADggEBAKt/7hwWqZw8UQCgwBEIBaeZ5m8BiFRh
bvG5GK1Krf6BQCOUL/t1fC8oS2IkgYIL9WHxHG64YTjrgfpioTtaYtOUZcTh5m2C
+C8lcLIhJsFyUR+MLMOEkMNaj7rP9KdlpeuY0fsFskZ1FSNqb4VjMIDw1Z4fKRzC
bLBQWV2QWzuoDTDPv31/zvGdg73JRm4gpvlhUbohL3u+pRVjodSVh/GeufOJ8z2F
uLjbvrW5KfnaNwUASZQDhETnv0Mxz3WLJdH0pmT1kvarBes96aULNmLazAZfNou2
XjG4Kvte9nHfRCaexOYNkbQudZWAUWpLMKawYqGT8ZvYzsRjdT9ZR7E=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDtTCCAp2gAwIBAgIQdrEgUnTwhYdGs/gjGvbCwDANBgkqhkiG9w0BAQsFADBt
MQswCQYDVQQGEwJDSDEQMA4GA1UEChMHV0lTZUtleTEiMCAGA1UECxMZT0lTVEUg
Rm91bmRhdGlvbiBFbmRvcnNlZDEoMCYGA1UEAxMfT0lTVEUgV0lTZUtleSBHbG9i
YWwgUm9vdCBHQiBDQTAeFw0xNDEyMDExNTAwMzJaFw0zOTEyMDExNTEwMzFaMG0x
CzAJBgNVBAYTAkNIMRAwDgYDVQQKEwdXSVNlS2V5MSIwIAYDVQQLExlPSVNURSBG
b3VuZGF0aW9uIEVuZG9yc2VkMSgwJgYDVQQDEx9PSVNURSBXSVNlS2V5IEdsb2Jh
bCBSb290IEdCIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2Be3
HEokKtaXscriHvt9OO+Y9bI5mE4nuBFde9IllIiCFSZqGzG7qFshISvYD06fWvGx
WuR51jIjK+FTzJlFXHtPrby/h0oLS5daqPZI7H17Dc0hBt+eFf1Biki3IPShehtX
1F1Q/7pn2COZH8g/497/b1t3sWtuuMlk9+HKQUYOKXHQuSP8yYFfTvdv37+ErXNk
u7dCjmn21HYdfp2nuFeKUWdy19SouJVUQHMD9ur06/4oQnc/nSMbsrY9gBQHTC5P
99UKFg29ZkM3fiNDecNAhvVMKdqOmq0NpQSHiB6F4+lT1ZvIiwNjeOvgGUpuuy9r
M2RYk61pv48b74JIxwIDAQABo1EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUw
AwEB/zAdBgNVHQ4EFgQUNQ/INmNe4qPs+TtmFc5RUuORmj0wEAYJKwYBBAGCNxUB
BAMCAQAwDQYJKoZIhvcNAQELBQADggEBAEBM+4eymYGQfp3FsLAmzYh7KzKNbrgh
cViXfa43FK8+5/ea4n32cZiZBKpDdHij40lhPnOMTZTg+XHEthYOU3gf1qKHLwI5
gSk8rxWYITD+KJAAjNHhy/peyP34EEY7onhCkRd0VQreUGdNZtGn//3ZwLWoo4rO
ZvUPQ82nK1d7Y0Zqqi5S2PTt4W2tKZB4SLrhI6qjiey1q5bAtEuiHZeeevJuQHHf

aPFlTc58Bd9TZaml8LGXBHAVRgOY1NK/VLSgWH1Sb9pWJmLU2NuJMW8c8CLC02Ic
Nc1MaRVUGpCY3useX8p3x8uOPUNpnJpY0CQ73xtAln41rYHHTnG6iBM=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICaTCCAe+gAwIBAgIQISpWDK7aDKtARb8roi066jAKBggqhkjOPQQDAzBtMQsw
CQYDVQQGEwJDSDEQMA4GA1UEChMHV0lTZUtleTEiMCAGA1UECxMZT0lTVEUgRm91
bmRhdGlvbiBFbmRvcnNlZDEoMCYGA1UEAxMfT0lTVEUgV0lTZUtleSBHbG9iYWwg
Um9vdCBHQyBDQTAeFw0xNzA1MDkwOTQ4MzRaFw00MjA1MDkwOTU4MzNaMG0xCzAJ
BgNVBAYTAkNIMRAwDgYDVQQKEwdXSVNlS2V5MSIwIAYDVQQLExlPSVNURSBGb3Vu
ZGF0aW9uIEVuZG9yc2VkMSgwJgYDVQQDEx9PSVNURSBXSVNlS2V5IEdsb2JhbCBS
b290IEdDIENBMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAETOlQwMYPchi82PG6s4ni
eUqjFqdrVCTbUf/q9Akkwwsin8tqJ4KBDdLArzHkdIJuyiXZjHWd8dvQmqJLIX4W
p2OQ0jnUsYd4XxiWD1AbNTcPasbc2RNNpI6QN+a9WzGRo1QwUjAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUSIcUrOPDnpBgOtfKie7T
rYy0UGYwEAYJKwYBBAGCNxUBBAMCAQAwCgYIKoZIzj0EAwMDaAAwZQIwJsdpW9zV
57LnyAyMjMPdeYwbY9XJUpROTYJKcx6ygISpJcBMWm1JKWB4E+J+SOtkAjEA2zQg
Mgj/mkkCtojeFK9dbJlxjRo/i9fgojaGHAeCOnZT/cKi7e97sIBPWA9LUzm9
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFYDCCA0igAwIBAgIUeFhfLq0sGUvjNwc1NBMotZbUZZMwDQYJKoZIhvcNAQEL
BQAwSDELMAkGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZhZGlzIExpbWl0ZWQxHjAc
BgNVBAMTFVF1b1ZhZGlzIFJvb3QgQ0EgMSBHMzAeFw0xMjAxMTIxNzI3NDRaFw00
MjAxMTIxNzI3NDRaMEgxCzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM
aW1pdGVkMR4wHAYDVQQDExVRdW9WYWRpcyBSb290IENBIDEgRzMwggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCgvlAQjunybEC0BJyFuTHK3C3kEakEPBtV
wedYMB0ktMPvhd6MLOHBPd+C5k+tR4ds7FtJwUrVu4/sh6x/gpqG7D0DmVIB0jWe
rNrwU8lmPNSsAgHaJNM7qAJGr6Qc4/hzWHa39g6QDbXwz8z6+cZM5cOGMAqNF341
68Xfuw6cwI2H44g4hWf6Pser4BOcBRiYz5P1sZK0/CPTz9XEJ0ngnjybCKOLXSoh
4Pw5qlPafX7PGglTvF0FBM+hSo+LdoINofjSxxR3W5A2B4GbPgb6Ul5jxaYA/qXp
UhtStZI5cgMJYr2wYBZupt0lwgNm3fME0UDiTouG9G/lg6AnhF4EwfWQvTA9xO+o
abw4m6SkltFi2mnAAZauy8RRNOoMqv8hjlmPSlzkYZqn0ukqeI1RPToV7qJZjqlc
3sX5kCLliEVx3ZGZbHqfPT2YfF72vhZooF6uCyP8Wg+qInYtyaEQHeTTRCOQiJ/G
KubX9ZqzWB4vMIkIG1SitZgj7Ah3HJVdYdHLiZxfokqRmu8hqkkWCKi9YSgxyXSt
hfbZxbGL0eUQMk1fiyA6PEkfM4VZDdvLCXVDaXP7a3F98N/ETH3Goy7IlXnLc6KO
Tk0k+17kBL5yG6YnLUlamXrXXAkgt3+UuU/xDRxeiEIbEbfnkduebPRq34wGmAOt
zCjvpUfzUwIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB
BjAdBgNVHQ4EFgQUo5fW816iEOGrRZ88F2Q87gFwnMwwDQYJKoZIhvcNAQELBQAD
ggIBABj6W3X8PnrHX3fHyt/PX8MSxEBd1DKquGrX1RUVRpgjpeaQWxiZTOOtQqOC
MTaIzen7xASWSIsBx40Bz1szBpZGZnQdT+3Btrm0DWHMY37XLneMlhwqI2hrhVd2
cDMT/uFPpiN3GPoajOi9ZcnPP/TJF9zrx7zABC4tRi9pZsMbj/7sPtPKlL92CiUN
qXsCHKnQO18LwIE6PWThv6ctTr1NxNgpxiIY0MWscgKCP6o6ojoilzHdCGPDdRS5
YCgtW2jgFqlmgiNR9etT2DGbe+m3nUvriBbP+V04ikkwj+3x6xn0dxoxGE1nVGwv
b2X52z3sIexe9PSLymBlVNFxZPT5pqOBMzYzcfCkeF9OrYMh3jRJjehZrJ3ydlo2
8hP0r+AJx2EqbPfgna67hkooby7utHnNkDPDs3b69fBsnQGQ+p6Q9pxyz0fawx/k

NSBT8lTR32GDpgLiJTjehTItXnOQUl1CxM49S+H5GYQd1aJQzEH7QRTDvdbJWqNj
ZgKAvQU6O0ec7AAmTPWIUb+oI38YB7AL7YsmoWTTYUrrXJ/es69nA7Mf3W1daWhp
q1467HxpvMc7hU6eFbm0FU/DlXpY18ls6Wy58yljXrQs8C097Vpl4KlbQMJImYFt
nh8GKjwStIsPm6Ik8KaN1nrgS7ZklmOVhMJKzRwuJIczYOXD
-----END
CERTIFICATE-----

-----BEGIN
CERTIFICATE-----
MIIFtzCCA5+gAwIBAgICBQkwDQYJKoZIhvcNAQEFBQAwRTELMAkGA1UEBhMCQk0x
GTAXBgNVBAoTEFF1b1ZhZGlzIExpbWl0ZWQxGzAZBgNVBAMTElF1b1ZhZGlzIFJv
b3QgQ0EgMjAeFw0wNjExMjQxODI3MDBaFw0zMTExMjQxODIzMzNaMEUxCzAJBgNV
BAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBMaW1pdGVkMRswGQYDVQQDExJRdW9W
YWRpcyBSb290IENBIDIwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCa
GMpLlA0ALa8DKYrwD4HIrkwZhR0In6spRIXzL4GtMh6QRr+jhiYaHv5+HBg6XJxg
Fyo6dIMzMH1hVBHL7avg5tKifvVrbxi3Cgst/ek+7wrGsxDp3MJGF/hd/aTa/55J
WpzmM+Yklvc/ulsrHHo1wtZn/qtmUIttKGAr79dgw8eTvI02kfN/+NsRE8Scd3bB
rrcCaoF6qUWD4gXmuVbBlDePSHFjIuwXZQeVikvfj8ZaCuWw419eaxGrDPmF60Tp
+ARz8un+XJiM9XOva7R+zdRcAitMOeGylZUtQofX1bOQQ7dsE/He3fbE+Ik/0XX1
ksOR1YqI0JDs3G3eicJlcZaLDQP9nL9bFqyS2+r+eXyt66/3FsvbzSUr5R/7mp/i
Ucw6UwxI5g69ybR2BlLmEROFcmMDBOAENisgGQLodKcftslWZvB1JdxnwQ5hYIiz
PtGo/KPaHbDRsSNU30R2be1B2MGyIrZTHN81Hdyhdyox5C315eXbyOD/5YDXC2Og
/zOhD7osFRXql7PSorW+8oyWHhqPHWykYTe5hnMz15eWniN9gqRMgeKh0bpnX5UH
oycR7hYQe7xFSkyyBNKr79X9DFHOUGoIMfmR2gyPZFwDwzqLID9ujWc9Otb+fVuI
yV77zGHcizN300QyNQliBJIWENieJ0f7OyHj+OsdWwIDAQABo4GwMIGtMA8GA1Ud
EwEB/wQFMAMBAf8wCwYDVR0PBAQDAgEGMB0GA1UdDgQWBBQahGK8SEwzJQTU7tD2
A8QZRtGUazBuBgNVHSMEZzBlgBQahGK8SEwzJQTU7tD2A8QZRtGUa6FJpEcwRTEL
MAkGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZhZGlzIExpbWl0ZWQxGzAZBgNVBAMT
ElF1b1ZhZGlzIFJvb3QgQ0EgMoICBQkwDQYJKoZIhvcNAQEFBQADggIBAD4KFk2f
BluornFdLwUvZ+YTRYPENvbzwCYMDbVHZF34tHLJRqUDGCdViXh9duqWNIAXINzn
g/iN/Ae42l9NLmeyhP3ZRPx3UIHmfLTJDQtyU/h2BwdBR5YM++CCJpNVjP4iH2Bl
fF/nJrP3MpCYUNQ3cVX2kiF495V5+vgtJodmVjB3pjd4M1IQWK4/YY7yarHvGH5K
WWPKjaJW1acvvFYfzznB4vsKqBUsfU16Y8Zsl0Q80m/DShcK+JDSV6IZUaUtl0Ha
B0+pUNqQjZRG4T7wlP0QADj1O+hA4bRuVhogzG9Yje0uRY/W6ZM/57Es3zrWIozc
hLsib9D45MY56QSIPMO661V6bYCZJPVsAfv4l7CUW+v90m/xd2gNNWQjrLhVoQPR
TUIZ3Ph1WVaj+ahJefivDrkRoHy3au000LYmYjgahwz46P0u05B/B5EqHdZ+XIWD
mbA4CD/pXvk1B+TJYm5Xf6dQlfe6yJvmjqIBxdZmv3lh8zwc4bmCXF2gw+nYSL0Z
ohEUGW6yhhtoPkg3Goi3XZZenMfvJ2II4pEZXNLxId26F0KCl3GBUzGpn/Z9Yr9y
4aOTHcyKJloJONDO1w2AFrR4pTqHTI2KpdVGl/IsELm8VCLAAVBpQ570su9t+Oza
8eOx79+Rj1QqCyXBJhnEUhAFZdWCEOrCMc0u
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFYDCCA0igAwIBAgIURFc0JFuBiZs18s64KztbpybwdSgwDQYJKoZIhvcNAQEL
BQAwSDELMAkGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZhZGlzIExpbWl0ZWQxHjAc
BgNVBAMTFVF1b1ZhZGlzIFJvb3QgQ0EgMiBHMzAeFw0xMjAxMTIxODU5MzJaFw00
MjAxMTIxODU5MzJaMEgxCzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM

aW1pdGVkMR4wHAYDVQQDExVRdW9WYWRpcyBSb290IENBIDIgRzMwggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQChriWyARjcV4g/Ruv5r+LrI3HimtFhZiFf
qq8nUeVuGxbULX1QsFN3vXg6YOJkApt8hpvWGo6t/x8Vf9WVHhLL5hSEBMHfNrMW
n4rjyduYNM7YMxcoRvynyfDStNVNCXJJ+fKH46nafaF9a7I6JaltUkSs+L5u+9ym
c5GQYaYDFCDy54ejiK2toIz/pgslUiXnFgHVy7g1gQyjO/Dh4fxaXc6AcW34Sas+
O7q414AB+6XrW7PFXmAqMaCvN+ggOp+oMiwMzAkd056OXbxMmO7FGmh77FOm6RQ1
o9/NgJ8MSPsc9PG/Srj61YxxSscfrf5BmrODXfKEVu+lV0POKa2Mq1W/xPtbAd0j
IaFYAI7D0GoT7RPjEiuA3GfmlbLNHiJuKvhB1PLKFAeNilUSxmn1uIZoL1NesNKq
IcGY5jDjZ1XHm26sGahVpkUG0CM62+tlXSoREfA7T8pt9DTEceT/AFr2XK4jYIVz
8eQQsSWu1ZK7E8EM4DnatDlXtas1qnIhO4M15zHfeiFuuDIIfR0ykRVKYnLP43eh
vNURG3YBZwjgQQvD6xVu+KQZ2aKrr+InUlYrAoosFCT5v0ICvybIxo/gbjh9Uy3l
7ZizlWNof/k19N+IxWA1ksB8aRxhlRbQ694Lrz4EEEVlWFA4r0jyWbYW8jwNkALG
cC4BrTwV1wIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB
BjAdBgNVHQ4EFgQU7edvdlq/YOxJW8ald7tyFnGbxD0wDQYJKoZIhvcNAQELBQAD
ggIBAJHfgD9DCX5xwvfrs4iP4VGyvD11+ShdyLyZm3tdquXK4Qr36LLTn91nMX66
AarHakE7kNQIXLJgapDwyM4DYvmL7ftuKtwGTTwpD4kWilhMSA/ohGHqPHKmd+RC
roijQ1h5fq7KpVMNqT1wvSAZYaRsOPxDMuHBR//47PERIjKWnML2W2mWeyAMQ0Ga
W/ZZGYjeVYg3UQt4XAoeo0L9x52ID8DyeAIkVJOviYeIyUqAHerQbj5hLja7NQ4n
lv1mNDthcnPxFlxHBlRJAHpYErAK74X9sbgzdWqTHBLmYF5vHX/JHyPLhGGfHoJE
+V+tYlUkmlKY7VHnoX6XOuYvHxHaU4AshZ6rNRDbIl9qxV6XU/IyAgkwo1jwDQHV
csaxfGl7w/U2Rcxhbl5MlMVerugOXou/983g7aEOGzPuVBj+D77vfoRrQ+NwmNtd
dbINWQeFFSM51vHfqSYP1kjHs6Yi9TM3WpVHn3u6GBVv/9YUZINJ0gpnIdsPNWNg
KCLjsZWDzYWm3S8P52dSbrsvhXz1SnPnxT7AvSESBT/8twNJAlvIJebiVDj1eYeM
HVOyToV7BjjHLPj4sHKNJeV3UvQDHEimUF+IIDBu8oJDqz2XhOdT+yHBTw8imoa4
WSr2Rz0ZiC3oheGe7IUIarFsNMkd7EgrO3jtZsSOeWmD3n+M
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIGnTCCBIWgAwIBAgICBcYwDQYJKoZIhvcNAQEFBQAwRTELMAkGA1UEBhMCQk0x
GTAXBgNVBAoTEFF1b1ZhZGlzIExpbWl0ZWQxGzAZBgNVBAMTElF1b1ZhZGlzIFJv
b3QgQ0EgMzAeFw0wNjExMjQxOTExMjNaFw0zMTExMjQxOTA2NDRaMEUxCzAJBgNV
BAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBMaW1pdGVkMRswGQYDVQQDExJRdW9W
YWRpcyBSb290IENBIDMwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDM
V0IWVJzmmNPTTe7+7cefQzlKZbPoFog02w1ZkXTPkrgEQK0CSzGrvI2RaNggDhoB
4hp7Thdd4oq3P5kazethq8Jlph+3t723j/z9cI8LoGe+AaJZz3HmDyl2/7FWeUUr
H556VOijKTVopAFPD6QuN+8bv+OPEKhyq1hX51SGyMnzW9os2l2ObjyjPtr7guXd
8lyyBTNvijbO0BNO/79KDDRMpsMhvVAEVeuxu537RR5kFd5VAYwCdrXLoT9Cabwv
vWhDFlaJKjdhkf2mrk7AyxRllDdLkgbvBNDInIjbC3uBr7E9KsRlOni27tyAsdLT
mZw67mtaa7ONt9XOnMK+pUsvFrGeaDsGb659n/je7Mwpp5ijJUMv7/FfJuGITfhe
btfZFG4ZM2mnO4SJk8RTVROhUXhA+LjJou57ulJCg54U7QVSWllWp5f8nT8KKdjc
T5EOE7zelaTfi5m+rJsziO+1ga8bxiJTyPbH7pcUsMV8eFLI8M5ud2CEpukqdiDt
WAEXMJPpGovgc2PZapKUSU60rUqFxKMiMPwJ7Wgic6aIDFUhWMXhOp8q3crhkODZ
c6tsgLjoC2SToJyMGf+z0gzskSaHirOi4XCPLArlzW1oUevaPwV/izLmE1xr/l9A
4iLItLRkT9a6fUg+qGkM17uGcclzuD87nSVL2v9A6wIDAQABo4IBlTCCAZEwDwYD
VR0TAQH/BAUwAwEB/zCB4QYDVR0gBIHZMIHWMIHTBgkrBgEEAb5YAAMwgcUwgZMG
CCsGAQUFBwICMIGGGoGDQW55IHVzZSBvZiB0aGlzIENlcnRpZmljYXRlIGNvbnN0
aXR1dGVzIGFjY2VwdGFuY2Ugb2YgdGhlIFF1b1ZhZGlzIFJvb3QgQ0EgMyBDZXJ0

aWZpY2F0ZSBQb2xpY3kgLyBDZXJ0aWZpY2F0aW9uIFByYWN0aWNlIFN0YXRlbWVu
dC4wLQYIKwYBBQUHAgEWIWh0dHA6Ly93d3cucXVvdmFkaXNnbG9iYWwuY29tL2Nw
czALBgNVHQ8EBAMCAQYwHQYDVR0OBBYEFPLAE+CCQz777i9nMpY1XNu4ywLQMG4G
A1UdIwRnMGWAFPLAE+CCQz777i9nMpY1XNu4ywLQoUmkRzBFMQswCQYDVQQGEwJC
TTEZMBcGA1UEChMQUXVvVmFkaXMgTGltaXRlZDEbMBkGA1UEAxMSUXVvVmFkaXMg
Um9vdCBDQSAzggIFxjANBgkqhkiG9w0BAQUFAAOCAgEAT62gLEz6wPJv92ZVqyM0
7ucp2sNbtrCD2dDQ4iH782CnO11gUyeim/YIIirnv6By5ZwkajGxkHon24QRiSem
d1o417+shvzuXYO8BsbRd2sPbSQvS3pspweWyuOEn62Iix2rFo1bZhfZFvSLgNLd
+LJ2w/w4E6oM3kJpK27zPOuAJ9v1pkQNn1pVWQvVDVJIxa6f8i+AxeoyUDUSly7B
4f/xI4hROJ/yZlZ25w9Rl6VSDE1JUZU2Pb+iSwwQHYaZTKrzchGT5Or2m9qoXadN
t54CrnMAyNojA+j56hl0YgCUyyIgvpSnWbWCar6ZeXqp8kokUvd0/bpO5qgdAm6x
DYBEwa7TIzdfu4V8K5Iu6H6li92Z4b8nby1dqnuH/grdS/yO9SbkbnBCbjPsMZ57
k8HkyWkaPcBrTiJt7qtYTcbQQcEr6k8Sh17rRdhs9ZgC06DYVYoGmRmioHfRMJ6s
zHXug/WwYjnPbFfiTNKRCw51KBuav/0aQ/HKd/s7j2G4aSgWQgRecCocIdiP4b0j
Wy10QJLZYxkNc91pvGJHvOB0K7Lrfb5BG7XARsWhIstfTsEokt4YutUqKLsRixeT
mJlglFwjz1onl14LBQaTNx47aTbrqZ5hHY8y2o4M1nQ+ewkk2gF3R8Q7zTSMmfXK
4SVhM7JZG+Ju1zdXtg2pEto=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFYDCCA0igAwIBAgIULvWbAiin23r/1aOp7r0DoM8Sah0wDQYJKoZIhvcNAQEL
BQAwSDELMAkGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZhZGlzIExpbWl0ZWQxHjAc
BgNVBAMTFVF1b1ZhZGlzIFJvb3QgQ0EgMyBHMzAeFw0xMjAxMTIyMDI2MzJaFw00
MjAxMTIyMDI2MzJaMEgxCzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM
aW1pdGVkMR4wHAYDVQQDExVRdW9WYWRpcyBSb290IENBIDMgRzMwggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCzyw4QZ47qFJenMioKVjZ/aEzHs286IxSR
/xl/pcqs7rN2nXrpixurazHb+gtTTK/FpRp5PIpM/6zfJd5O2YIyC0TeytuMrKNu
FoM7pmRLMon7FhY4futD4tN0SsJiCnMK3UmzV9KwCoWdcTzeo8vAMvMBOSBDGzXR
U7Ox7sWTaYI+FrUoRqHe6okJ7UO4BUaKhvVZR74bbwEhELn9qdIoyhA5CcoTNs+c
ra1AdHkrAj80//ogaX3T7mH1urPnMNA3I4ZyYUUpSFlob3emLoG+B01vr87ERROR
FHAGjx+f+IdpsQ7vw4kZ6+ocYfx6bIrc1gMLnia6Et3UVDmrJqMz6nWB2i3ND0/k
A9HvFZcba5DFApCTZgIhsUfei5pKgLlVj7WiL8DWM2fafsSntARE60f75li59wzw
eyuxwHApw0BiLTtIadwjPEjrewl5qW3aqDCYz4ByA4imW0aucnl8CAMhZa634Ryl
sSqiMd5mBPfAdOhx3v89WcyWJhKLhZVXGqtrdQtEPREoPHtht+KPZ0/l7DxMYIBp
VzgeAVuNVejH38DMdyM0SXV89pgR6y3e7UEuFAUCf+D+IOs15xGsIs5XPd7JMG0Q
A4XN8f+MFrXBsj6IbGB/kE+V9/YtrQE5BwT6dYB9v0lQ7e/JxHwc64B+27bQ3RP+
ydOc17KXqQIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB
BjAdBgNVHQ4EFgQUxhfQvKjqAkPyGwaZXSuQILnXnOQwDQYJKoZIhvcNAQELBQAD
ggIBADRh2Va1EodVTd2jNTFGu6QHcrxfYWLopfsLN7E8trP6KZ1/AvWkyaiTt3px
KGmPc+FSkNrVvjrlt3ZqVoAh313m6Tqe5T72omnHKgqwGEfcIHB9UqM+WXzBusnI
FUBhynLWcKzSt/Ac5IYp8M7vaGPQtSCKFWGafoaYtMnCdvvMujAWzKNhxnQT5Wvv
oxXqA/4Ti2Tk08HS6IT7SdEQTXlm66r99I0xHnAUrdzeZxNMgRVhvLfZkXdxGYFg
u/BYpbWcC/ePIlUnwEsBbTuZDdQdm2NnL9DuDcpmvJRPpq3t/O5jrFc/ZSXPsoaP
0Aj/uHYUbt7lJ+yreLVTubY/6CD50qi+YUbKh4yE8/nxoGibIh6BJpsQBJFxwAYf
3KDTuVan45gtf4Od34wrnDKOMpTwATwiKp9Dwi7DmDkHOHv8XgBCH/MyJnmDhPbl
8MFREsALHgQjDFSlTC9JxUrRtm5gDWv8a4uFJGS3iQ6rJUdbPM9+Sb3H6QrG2vd+
DhcI00iX0HGS8A85PjRqHH3Y8iKuu2n0M7SmSFXRDw4m6Oy2Cy2nhTXN/VnIn9HN

PlopNLk9hM6xZdRZkZFWdSHBd575euFgndOtBBj0fOtek49TSiIp+EgrPk2GrFt/
ywaZWWDYWGWVjUTR939+J399roD1B0y2PpxxVJkES/1Y+Zj0
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIClDCCAhqgAwIBAgIILCmcWxbtBZUwCgYIKoZIzj0EAwIwfzELMAkGA1UEBhMC
VVMxDjAMBgNVBAgMBVRleGFzMRAwDgYDVQQHDAdIb3VzdG9uMRgwFgYDVQQKDA9T
U0wgQ29ycG9yYXRpb24xNDAyBgNVBAMMK1NTTC5jb20gRVYgUm9vdCBDZXJ0aWZp
Y2F0aW9uIEF1dGhvcml0eSBFQ0MwHhcNMTYwMjEyMTgxNTIzWhcNNDEwMjEyMTgx
NTIzWjB/MQswCQYDVQQGEwJVUzEOMAwGA1UECAwFVGV4YXMxEDAOBgNVBAcMB0hv
dXN0b24xGDAWBgNVBAoMD1NTTCBDb3Jwb3JhdGlvbjE0MDIGA1UEAwwrU1NMLmNv
bSBFViBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5IEVDQzB2MBAGByqGSM49
AgEGBSuBBAAiA2IABKoSR5CYG/vvw0AHgyBO8TCCogbR8pKGYfL2IWjKAMTH6kMA
VIbc/R/fALhBYlzccBYy3h+Z1MzFB8gIH2EWB1E9fVwHU+M1OIzfzZ/ZLg1Kthku
WnBaBu2+8KGwytAJKaNjMGEwHQYDVR0OBBYEFFvKXuXe0oGqzagtZFG22XKbl+ZP
MA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUW8pe5d7SgarNqC1kUbbZcpuX
5k8wDgYDVR0PAQH/BAQDAgGGMAoGCCqGSM49BAMCA2gAMGUCMQCK5kCJN+vp1RPZ
ytRrJPOwPYdGWBrssd9v+1a6cGvHOMzosYxPD/fxZ3YOg9AeUY8CMD32IygmTMZg
h5Mmm7I1HrrW9zzRHM76JTymGoEVW/MSD2zuZYrJh6j5B+BimoxcSg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF6zCCA9OgAwIBAgIIVrYpzTS8ePYwDQYJKoZIhvcNAQELBQAwgYIxCzAJBgNV
BAYTAlVTMQ4wDAYDVQQIDAVUZXhhczEQMA4GA1UEBwwHSG91c3RvbjEYMBYGA1UE
CgwPU1NMIENvcnBvcmF0aW9uMTcwNQYDVQQDDC5TU0wuY29tIEVWIFJvb3QgQ2Vy
dGlmaWNhdGlvbiBBdXRob3JpdHkgUlNBIFIyMB4XDTE3MDUzMTE4MTQzN1oXDTQy
MDUzMDE4MTQzN1owgYIxCzAJBgNVBAYTAlVTMQ4wDAYDVQQIDAVUZXhhczEQMA4G
A1UEBwwHSG91c3RvbjEYMBYGA1UECgwPU1NMIENvcnBvcmF0aW9uMTcwNQYDVQQD
DC5TU0wuY29tIEVWIFJvb3QgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkgUlNBIFIy
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAjzZlQOHWTcDXtOlG2mvq
M0fNTPl9fb69LT3w23jhhqXZuglXaO1XPqDQCEGD5yhBJB/jchXQARr7XnAjssuf
OePPxU7Gkm0mxnu7s9onnQqG6YE3Bf7wcXHswxzpY6IXFJ3vG2fThVUCAtZJycxa
4bH3bzKfydQ7iEGonL3Lq9ttewkfokxykNorCPzPPFTOZw+oz12WGQvE43LrrdF9
HSfvkusQv1vrO6/PgN3B0pYEW3p+pKk8OHakYo6gOV7qd89dAFmPZiw+B6KjBSYR
aZfqhbcPlgtLyEDhULouisv3D5oi53+aNxPN8k0TayHRwMwi8qFG9kRpnMphNQcA
b9ZhCBHqurj26bNg5U257J8UZslXWNvNh2n4ioYSA0e/ZhN2rHd9NCSFg83XqpyQ
Gp8hLH94t2S42Oim9HizVcuE0jLEeK6jj2HdzghTreyI/BXkmg3mnxp3zkyPuBQV
PWKchjgGAGYS5Fl2WlPAApiiECtoRHuOec4zSnaqW4EWG7WK2NAAe15itAnWhmMO
pgWVSbooi4iTsjQc2KRVbrcc0N6ZVTsj9CLg+SlmJuwgUHfbSguPvuUCYHBBXtSu
UDkiFCbLsjtzdFVHB3mBOagwE0TlBIqulhMlQg+5U8Sb/M3kHN48+qvWBkofZ6aY
MBzdLNvcGJVXZsb/XItW9XcCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAfBgNV
HSMEGDAWgBT5YLvU49U09rj1BoAlp3PbRmmonjAdBgNVHQ4EFgQU+WC71OPVNPa4
9QaAJadz20ZpqJ4wDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3DQEBCwUAA4ICAQBW
s47LCp1Jjr+kxJG7ZhcFUZh1++VQLHqe8RT6q9OKPv+RKY9ji9i0qVQBDb6Thi/5
Sm3HXvVX+cpVHBK+Rw82xd9qt9t1wkclf7nxY/hoLVUE0fKNsKTPvDxeH3jnpaAg
cLAExbf3cqfeIg29MyVGjGSSJuM+LmOW2puMPfgYCdcDzH2GguDKBAdRUNf/ktUM

79qGn5nX67evaOI5JpS6aLe/g9Pqemc9YmeuJeVy6OLk7K4S9ksrPJ/psEDzOFSz
/bdoyNrGj1E8svuR3Bznm53htw1yj+KkxKl4+esUrMZDBcJlOSgYAsOCsp0FvmXt
ll9ldDz7CTUue5wT/RsPXcdtgTpWD8w74a8CLyKsRspGPKAcTNZEtF4uXBVmCeEm
Kf7GUmG6sXP/wwyc5WxqlD8UykAWlYTzWamsX0xhk23RO8yilQwipmdnRC652dKK
QbNmC1r7fSOl8hqw/96bg5Qu0T/fkreRrwU7ZcegbLHNYhLDkBvjJc40vG93drEQ
w/cFGsDWr3RiSBd3kmmQYRzelYB0VI8YHMPzA9C/pEN1hlMYegouCRw2n5H9gooi
S9EOUCXdywMMF8mDAAhONU2Ki+3wApRmLER/y5UnlhetCTCstnEXbosX9hwJ1C07
mKVx01QT2WDz9UtmT/rx7iASjbSsV7FFY6GsdqnC+w==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICjTCCAhSgAwIBAgIIdebfy8FoW6gwCgYIKoZIzj0EAwIwfDELMAkGA1UEBhMC
VVMxDjAMBgNVBAgMBVRleGFzMRAwDgYDVQQHDAdIb3VzdG9uMRgwFgYDVQQKDA9T
U0wgQ29ycG9yYXRpb24xMTAvBgNVBAMMKFNTTC5jb20gUm9vdCBDZXJ0aWZpY2F0
aW9uIEF1dGhvcml0eSBFQ0MwHhcNMTYwMjEyMTgxNDAzWhcNNDEwMjEyMTgxNDAz
WjB8MQswCQYDVQQGEwJVUzEOMAwGA1UECAwFVGV4YXMxEDAOBgNVBAcMB0hvdXN0
b24xGDAWBgNVBAoMD1NTTCBDb3Jwb3JhdGlvbjExMC8GA1UEAwwoU1NMLmNvbSBS
b290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5IEVDQzB2MBAGByqGSM49AgEGBSuB
BAAiA2IABEVuqVDEpiM2nl8ojRfLliJkP9x6jh3MCLOicSS6jkm5BBtHllirLZXI
7Z4INcgn64mMU1jrYor+8FsPazFSY0E7ic3s7LaNGdM0B9y7xgZ/wkWV7Mt/qCPg
CemB+vNH06NjMGEwHQYDVR0OBBYEFILRhXMw5zUE044CkvvlpNHEIejNMA8GA1Ud
EwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUgtGFczDnNQTTjgKS++Wk0cQh6M0wDgYD
VR0PAQH/BAQDAgGGMAoGCCqGSM49BAMCA2cAMGQCMG/n61kRpGDPYbCWe+0F+S8T
kdzt5fxQaxFGRrMcIQBiu77D5+jNB5n5DQtdcj7EqgIwH7y6C+IwJPt8bYBVCpk+
gA0z5Wajs6O7pdWLjwkspl1+4vAHCGht0nxpbl/f5Wpl
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF3TCCA8WgAwIBAgIIeyyb0xaAMpkwDQYJKoZIhvcNAQELBQAwfDELMAkGA1UE
BhMCVVMxDjAMBgNVBAgMBVRleGFzMRAwDgYDVQQHDAdIb3VzdG9uMRgwFgYDVQQK
DA9TU0wgQ29ycG9yYXRpb24xMTAvBgNVBAMMKFNTTC5jb20gUm9vdCBDZXJ0aWZp
Y2F0aW9uIEF1dGhvcml0eSBSU0EwHhcNMTYwMjEyMTczOTM5WhcNNDEwMjEyMTcz
OTM5WjB8MQswCQYDVQQGEwJVUzEOMAwGA1UECAwFVGV4YXMxEDAOBgNVBAcMB0hv
dXN0b24xGDAWBgNVBAoMD1NTTCBDb3Jwb3JhdGlvbjExMC8GA1UEAwwoU1NMLmNv
bSBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5IFJTQTCCAiIwDQYJKoZIhvcN
AQEBBQADggIPADCCAgoCggIBAPkP3aMrfcvQKv7sZ4Wm5y4bunfh4/WvpOz6Sl2R
xFdHaxh3a3by/ZPkPQ/CFp4LZsNWlJ4Xg4XOVu/yFv0AYvUiCVToZRdOQbngT0aX
qhvIuG5iXmmxX9sqAn78bMrzQdjt0Oj8P2FI7bADFB0QDksZ4LtO7IZl/zbzXmcC
C52GVWH9ejjt/uIZALdvoVBidXQ8oPrIJZK0bnoix/geoeOy3ZExqysdBP+lSgQ3
6YWkMyv94tZVNHwZpEpox7Ko07fKoZOI68GXvIz5HdkihCR0xwQ9aqkpk8zruFvh
/l8lqjRYyMEjVJ0bmBHDOJx+PYZspQ9AhnwC9FwCTyjLrnGfDzrIM/4RJTXq/LrF
YD3ZfBjVsqnTdXgDciLKOsMf7yzlLqn6niy2UUb9rwPW6mBo6oUWNmuF6R7As93E
JNyAKoFBbZQ+yODJgUEAnl6/f8UImKIYLEJAs/lvOCdLToD0PYFH4Ih86hzOtXVc
US4cK38acijnALXRdMbX5J+tB5O2UzU1/Dfkw/ZdFr4hc96SCvigY2q8lpJqPvi8
ZVWb3vUNiSYE/CUapiVpy8JtynziWV+XrOvvLsi81xtZPCvM8hnIk2snYxnP/Okm
+Mpxm3+T/jRnhE6Z6/yzeAkzcLpmpnbtG3PrGqUNxCITIJRWCk4sbE6x/c+cCbqi

M+2HAgMBAAGjYzBhMB0GA1UdDgQWBBTdBAkHovV6fVJTEpKV7jiAJQ2mWTAPBgNV
HRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFN0ECQei9Xp9UlMSkpXuOIAlDaZZMA4G
A1UdDwEB/wQEAwIBhjANBgkqhkiG9w0BAQsFAAOCAgEAIBgRlCn7Jp0cHh5wYfGV
cpNxJK1ok1iOMq8bs3AD/CUrdIWQPXhq9LmLpZc7tRiRux6n+UBbkflVma8eEdBc
Hadm47GUBwwyOabqG7B52B2ccETjit3E+ZUfijhDPwGFpUenPUayvOUiaPd7nNgs
PgohyC0zrL/FgZkxdMF1ccW+sfAjRfSda/wZY52jvATGGAslu1OJD7OAUN5F7kR/
q5R4ZJjT9ijdh9hwZXT7DrkT66cPYakylszeu+1jTBi7qUD3oFRuIIhxdRjqerQ0
cuAjJ3dctpDqhiVAq+8zD8ufgr6iIPv2tS0a5sKFsXQP+8hlAqRSAUfdSSLBv9jr
a6x+3uxjMxW3IwiPxg+NQVrdjsW5j+VFP3jbutIbQLH+cU0/4IGiul607BXgk90I
H37hVZkLId6Tngr75qNJvTYw/ud3sqB1l7UtgYgXZSD32pAAn8lSzDLKNXz1PQ/Y
K9f1JmzJBjSWFupwWRoyeXkLtoh/D1JIPb9s2KJELtFOt3JY04kTlf5Eq/jXixtu
nLwsoFvVagCvXzfh1foQC5ichucmj87w7G6KVwuA406ywKBjYZC6VWg3dGq2ktuf
oYYitmUnDuy2n0Jg5GfCtdpBC8TTi2EbvPofkSvXRAdeuims2cXp71NIWuuA8ShY
Ic2wBlX7Jz9TkHCpBB5XJ7k=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICOjCCAcCgAwIBAgIQFAP1q/s3ixdAW+JDsqXRxDAKBggqhkjOPQQDAzBOMQsw
CQYDVQQGEwJVUzEYMBYGA1UECgwPU1NMIENvcnBvcmF0aW9uMSUwIwYDVQQDDBxT
U0wuY29tIFRMUyBFQ0MgUm9vdCBDQSAyMDIyMB4XDTIyMDgyNTE2MzM0OFoXDTQ2
MDgxOTE2MzM0N1owTjELMAkGA1UEBhMCVVMxGDAWBgNVBAoMD1NTTCBDb3Jwb3Jh
dGlvbjElMCMGA1UEAwwcU1NMLmNvbSBUTFMgRUNDIFJvb3QgQ0EgMjAyMjB2MBAG
ByqGSM49AgEGBSuBBAAiA2IABEUpNXP6wrgjzhR9qLFNoFs27iosU8NgCTWyJGYm
acCzldZdkkAZDsalE3D07xJRKF3nzL35PIXBz5SQySvOkkJYWWf9lCcQZIxPBLFN
SeR7T5v15wj4A4j3p8OSSxlUgaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAfBgNVHSME
GDAWgBSJjy+j6CugFFR781a4Jl9nOAuc0DAdBgNVHQ4EFgQUiY8vo+groBRUe/NW
uCZfZzgLnNAwDgYDVR0PAQH/BAQDAgGGMAoGCCqGSM49BAMDA2gAMGUCMFXjIlbp
15IkWE8elDIPDAI2wv2sdDJO4fscgIijzPvX6yv/N33w7deedWo1dlJF4AIxAMeN
b0Igj762TVntd00pxCAgRWSGOlDGxK0tk/UYfXLtqc/ErFc2KAhl3zx5Zn6g6g==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFiTCCA3GgAwIBAgIQb77arXO9CEDii02+1PdbkTANBgkqhkiG9w0BAQsFADBO
MQswCQYDVQQGEwJVUzEYMBYGA1UECgwPU1NMIENvcnBvcmF0aW9uMSUwIwYDVQQD
DBxTU0wuY29tIFRMUyBSU0EgUm9vdCBDQSAyMDIyMB4XDTIyMDgyNTE2MzQyMloX
DTQ2MDgxOTE2MzQyMVowTjELMAkGA1UEBhMCVVMxGDAWBgNVBAoMD1NTTCBDb3Jw
b3JhdGlvbjElMCMGA1UEAwwcU1NMLmNvbSBUTFMgUlNBIFJvb3QgQ0EgMjAyMjCC
AiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBANCkCXJPQIgSYT41I57u9nTP
L3tYPc48DRAokC+X94xI2KDYJbFMsBFMF3NQ0CJKY7uB0ylu1bUJPiYYf7ISf5OY
t6/wNr/y7hienDtSxUcZXXTzZGbVXcdotL8bHAajvI9AI7YexoS9UcQbOcGV0ins
S657Lb85/bRi3pZ7QcacoOAGcvvwB5cJOYF0r/c0WRFXCsJbwST0MXMwgsadugL3
PnxEX4MN8/HdIGkWCVDi1FW24IBydm5MR7d1VVm0U3TZlMZBrViKMWYPHqIbKUBO
L9975hYsLfy/7PO0+r4Y9ptJ1O4Fbtk085zx7AGL0SDGD6C1vBdOSHtRwvzpXGk3
R2azaPgVKPC506QVzFpPulJwoxJF3ca6TvvC0PeoUidtbnm1jPx7jMEWTO6Af77w
dr5BUxIzrlo4QqvXDz5BjXYHMtWrifZOZ9mxQnUjbvPNQrL8VfVThxc7wDNY8VLS
+YCk8OjwO4s4zKTGkH8PnP2L0aPP2oOnaclQNtVcBdIKQXTbYxE3waWglksejBYS

d66UNHsef8JmAOSqg+qKkK3ONkRN0VHpvB/zagX9wHQfJRlAUW7qglFA35u5CCoG
AtUjHBPW6dvbxrB6y3snm/vg1UYk7RBLY0ulBY+6uB0rpvqR4pJSvezrZ5dtmi2f
gTIFZzL7SAg/2SW4BCUvAgMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0j
BBgwFoAU+y437uOEeicuzRk1sTN8/9REQrkwHQYDVR0OBBYEFPsuN+7jhHonLs0Z
NbEzfP/UREK5MA4GA1UdDwEB/wQEAwIBhjANBgkqhkiG9w0BAQsFAAOCAgEAjYlt
hEUY8U+zoO9opMAdrDC8Z2awms22qyIZZtM7QbUQnRC6cm4pJCAcAZli05bg4vsM
QtfhWsSWTVTNj8pDU/0quOr4ZcoBwq1gaAafORpR2eCNJvkLTqVTJXojpBzOCBvf
R4iyrT7gJ4eLSYwfqUdYe5byiB0YrrPRpgqU+tvT5TgKa3kSM/tKWTcWQA673vWJ
DPFs0/dRa1419dvAJuoSc06pkZCmF8NsLzjUo3KUQyxi4U5cMj29TH0ZR6LDSeeW
P4+a0zvkEdiLA9z2tmBVGKaBUfPhqBVq6+AL8BQx1rmMRTqoENjwuSfr98t67wVy
lrXEj5ZzxOhWc5y8aVFjvO9nHEMaX3cZHxj4HCUp+UmZKbaSPaKDN7EgkaibMOlq
bLQjk2UEqxHzDh1TJElTHaE/nUiSEeJ9DU/1172iWD54nR4fK/4huxoTtrEoZP2w
AgDHbICivRZQIA9ygV/MlP+7mea6kMvq+cYMwq7FGc4zoWtcu358NFcXrfA/rs3q
r5nsLFR+jM4uElZI7xc7P0peYNLcdDa8pUNjyw9bowJWCZ4kLOGGgYz+qxcs+sji
Mho6/4UIyYOf8kpIEFR3N+2ivEC+5BB09+Rbu7nzifmPQdjH5FCQNYA+HLhNkNPU
98OwoX6EyneSMSy4kLGCenROmxMmtNVQZlR4rmA=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDcjCCAlqgAwIBAgIUPopdB+xV0jLVt+O2XwHrLdzk1uQwDQYJKoZIhvcNAQEL
BQAwUTELMAkGA1UEBhMCUEwxKDAmBgNVBAoMH0tyYWpvd2EgSXpiYSBSb3psaWN6
ZW5pb3dhIFMuQS4xGDAWBgNVBAMMD1NaQUZJUiBST09UIENBMjAeFw0xNTEwMTkw
NzQzMzBaFw0zNTEwMTkwNzQzMzBaMFExCzAJBgNVBAYTAlBMMSgwJgYDVQQKDB9L
cmFqb3dhIEl6YmEgUm96bGljemVuaW93YSBTLkEuMRgwFgYDVQQDDA9TWkFGSVIg
Uk9PVCBDQTIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC3vD5QqEvN
QLXOYeeWyrSh2gwisPq1e3YAd4wLz32ohswmUeQgPYUM1ljj5/QqGJ3a0a4m7utT
3PSQ1hNKDJA8w/Ta0o4NkjrcsbH/ON7Dui1fgLkCvUqdGw+0w8LBZwPd3BucPbOw
3gAeqDRHu5rr/gsUvTaE2g0gv/pby6kWIK05YO4vdbbnl5z5Pv1+TW9NL++IDWr6
3fE9biCloBK0TXC5ztdyO4mTp4CEHCdJckm1/zuVnsHMyAHs6A6KCpbns6aH5db5
BSsNl0BwPLqsdVqc1U2dAgrSS5tmS0YHF2Wtn2yIANwiieDhZNRnvDF5YTy7ykHN
XGoAyDw4jlivAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQD
AgEGMB0GA1UdDgQWBBQuFqlKGLXLzPVvUPMjX/hd56zwyDANBgkqhkiG9w0BAQsF
AAOCAQEAtXP4A9xZWx126aMqe5Aosk3AM0+qmrHUuOQn/6mWmc5G4G18TKI4pAZw
8PRBEew/R40/cof5O/2kbytTAOD/OblqBw7rHRz2onKQy4I9EYKL0rufKq8h5mOG
nXkZ7/e7DDWQw4rtTw/1zBLZpD67oPwglV9PJi8RI4NOdQcPv5vRtB3pEAT+ymCP
oky4rc/hkA/NrgrHXXu3UNLUYfrVFdvXn4dRVOul4+vJhaAlIDf7js4MNIThPIGy
d05DpYhfhmehPea0XGG2Ptv+tyjFogeutcrKjSoS75ftwjCkySp6+/NNIxuZMzSg
LvWpCz/UXeHPhJ/iGcJfitYgHuNztw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICOjCCAcGgAwIBAgIQQvLM2htpN0RfFf51KBC49DAKBggqhkjOPQQDAzBfMQsw
CQYDVQQGEwJHQjEYMBYGA1UEChMPU2VjdGlnbyBMaW1pdGVkMTYwNAYDVQQDEy1T
ZWN0aWdvIFB1YmxpYyBTZXJ2ZXIgQXV0aGVudGljYXRpb24gUm9vdCBFNDYwHhcN
MjEwMzIyMDAwMDAwWhcNNDYwMzIxMjM1OTU5WjBfMQswCQYDVQQGEwJHQjEYMBYG
A1UEChMPU2VjdGlnbyBMaW1pdGVkMTYwNAYDVQQDEy1TZWN0aWdvIFB1YmxpYyBT

ZXJ2ZXIgQXV0aGVudGljYXRpb24gUm9vdCBFNDYwdjAQBgcqhkjOPQIBBgUrgQQA
IgNiAAR2+pmpbiDt+dd34wc7qNs9Xzjoq1WmVk/WSOrsfy2qw7LFeeyZYX8QeccC
WvkEN/U0NSt3zn8gj1KjAIns1aeibVvjS5KToID1AZTc8GgHHs3u/iVStSBDHBv+
6xnOQ6OjQjBAMB0GA1UdDgQWBBTRItpMWfFLXyY4qp3W7usNw/upYTAOBgNVHQ8B
Af8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkjOPQQDAwNnADBkAjAn7qRa
qCG76UeXlImldCBteU/IvZNeWBj7LRoAasm4PdCkT0RHlAFWovgzJQxC36oCMB3q
4S6ILuH5px0CMk7yn2xVdOOurvulGu7t0vzCAxHrRVxgED1cf5kDW21USAGKcw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFijCCA3KgAwIBAgIQdY39i658BwD6qSWn4cetFDANBgkqhkiG9w0BAQwFADBf
MQswCQYDVQQGEwJHQjEYMBYGA1UEChMPU2VjdGlnbyBMaW1pdGVkMTYwNAYDVQQD
Ey1TZWN0aWdvIFB1YmxpYyBTZXJ2ZXIgQXV0aGVudGljYXRpb24gUm9vdCBSNDYw
HhcNMjEwMzIyMDAwMDAwWhcNNDYwMzIxMjM1OTU5WjBfMQswCQYDVQQGEwJHQjEY
MBYGA1UEChMPU2VjdGlnbyBMaW1pdGVkMTYwNAYDVQQDEy1TZWN0aWdvIFB1Ymxp
YyBTZXJ2ZXIgQXV0aGVudGljYXRpb24gUm9vdCBSNDYwggIiMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCTvtU2UnXYASOgHEdCSe5jtrch/cSV1UgrJnwUUxDa
ef0rty2k1Cz66jLdScK5vQ9IPXtamFSvnl0xdE8H/FAh3aTPaE8bEmNtJZlMKpnz
SDBh+oF8HqcIStw+KxwfGExxqjWMrfhu6DtK2eWUAtaJhBOqbchPM8xQljeSM9xf
iOefVNlI8JhD1mb9nxc4Q8UBUQvX4yMPFF1bFOdLvt30yNoDN9HWOaEhUTCDsG3X
ME6WW5HwcCSrv0WBZEMNvSE6Lzzpng3LILVCJ8zab5vuZDCQOc2TZYEhMbUjUDM3
IuM47fgxMMxF/mL50V0yeUKH32rMVhlATc6qu/m1dkmU8Sf4kaWD5QazYw6A3OAS
VYCmO2a0OYctyPDQ0RTp5A1NDvZdV3LFOxxHVp3i1fuBYYzMTYCQNFu31xR13NgE
SJ/AwSiItOkcyqex8Va3e0lMWeUgFaiEAin6OJRpmkkGj80feRQXEgyDet4fsZfu
+Zd4KKTIRJLpfSYFplhym3kT2BFfrsU4YjRosoYwjviQYZ4ybPUHNs2iTG7sijbt
8uaZFURww3y8nDnAtOFr94MlI1fZEoDlSfB1D++N6xybVCi0ITz8fAr/73trdf+L
HaAZBav6+CuBQug4urv7qv094PPK306Xlynt8xhW6aWWrL3DkJiy4Pmi1KZHQ3xt
zwIDAQABo0IwQDAdBgNVHQ4EFgQUVnNYZJX5khqwEioEYnmhQBWIIUkwDgYDVR0P
AQH/BAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEMBQADggIBAC9c
mTz8Bl6MlC5w6tIyMY208FHVvArzZJ8HXtXBc2hkeqK5Duj5XYUtqDdFqij0lgVQ
YKlJfp/imTYpE0RHap1VIDzYm/EDMrraQKFz6oOht0SmDpkBm+S8f74TlH7Kph52
gDY9hAaLMyZlbcp+nv4fjFg4exqDsQ+8FxG75gbMY/qB8oFM2gsQa6H61SilzwZA
Fv97fRheORKkU55+MkIQpiGRqRxOF3yEvJ+M0ejf5lG5Nkc/kLnHvALcWxxPDkjB
JYOcCj+esQMzEhonrPcibCTRAUH4WAP+JWgiH5paPHxsnnVI84HxZmduTILA7rpX
DhjvLpr3Etiga+kFpaHpaPi8TD8SHkXoUsCjvxInebnMMTzD9joiFgOgyY9mpFui
TdaBJQbpdqQACj7LzTWb4OE4y2BThihCQRxEV+ioratF4yUQvNs+ZUH7G6aXD+u5
dHn5HrwdVw1Hr8Mvn4dGp+smWg9WY7ViYG4A++MnESLn/pmPNPW56MORcr3Ywx65
LvKRRFHQV80MNNVIIb/bE/FmJUNS0nAiNs2fxBx1IK1jcmMGDw4nztJqDby1ORrp
0XZ60Vzk50lJLVU3aPAaOpg+VBeHVOmmJ1CJeyAvP/+/oYtKR5j/K3tJPsMpRmAY
QqszKbrAKbkTidOIijlBO8n9pu0f9GBj39ItVQGL
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDcjCCAlqgAwIBAgIUZvnHwa/swlG07VOX5uaCwysckBYwDQYJKoZIhvcNAQEL
BQAwUTELMAkGA1UEBhMCSlAxIzAhBgNVBAoTGkN5YmVydHJ1c3QgSmFwYW4gQ28u
LCBMdGQuMR0wGwYDVQQDExRTZWN1cmVTaWduIFJvb3QgQ0ExMjAeFw0yMDA0MDgw

NTM2NDZaFw00MDA0MDgwNTM2NDZaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKExpD
eWJlcnRydXN0IEphcGFuIENvLiwgTHRkLjEdMBsGA1UEAxMUU2VjdXJlU2lnbiBS
b290IENBMTIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC6OcE3emhF
KxS06+QT61d1I02PJC0W6K6OyX2kVzsqdiUzg2zqMoqUm048luT9Ub+ZyZN+v/mt
p7JIKwccJ/VMvHASd6SFVLX9kHrko+RRWAPNEHl57muTH2SOa2SroxPjcf59q5zd
J1M3s6oYwlkm7Fsf0uZlfO+TvdhYXAvA42VvPMfKWeP+bl+sg779XSVOKik71gur
FzJ4pOE+lEa+Ym6b3kaosRbnhW70CEBFEaCeVESE99g2zvVQR9wsMJvuwPWW0v4J
hscGWa5Pro4RmHvzC1KqYiaqId+OJTN5lxZJjfU+1UefNzFJM3IFTQy2VYzxV4+K
h9GtxRESOaCtAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQD
AgEGMB0GA1UdDgQWBBRXNPN0zwRL1SXm8UC2LEzZLemgrTANBgkqhkiG9w0BAQsF
AAOCAQEAPrvbFxbS8hQBICw4g0utvsqFepq2m2um4fylOqyttCg6r9cBg0krY6Ld
mmQOmFxv3Y67ilQiLUoT865AQ9tPkbeGGuwAtEGBpE/6aouIs3YIcipJQMPTw4WJ
mBClnW8Zt7vPemVV2zfrPIpyMpcemik+rY3moxtt9XUa5rBouVui7mlHJzWhhpmA
8zNL4WukJsPvdFlseqJkth5Ew1DgDzk9qTPxpfPSvWKErI4cqc1avTc7bgoitPQV
55FYxTpE05Uo2cBl6XLK0A+9H7MV2anjpEcJnuDLN/v9vZfVvhgaaaI5gdka9at/
yOPiZwud9AzqVN/Ssq+xIvEg37xEHA==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFcjCCA1qgAwIBAgIUZNtaDCBO6Ncpd8hQJ6JaJ90t8sswDQYJKoZIhvcNAQEM
BQAwUTELMAkGA1UEBhMCSlAxIzAhBgNVBAoTGkN5YmVydHJ1c3QgSmFwYW4gQ28u
LCBMdGQuMR0wGwYDVQQDExRTZWN1cmVTaWduIFJvb3QgQ0ExNDAeFw0yMDA0MDgw
NzA2MTlaFw00NTA0MDgwNzA2MTlaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKExpD
eWJlcnRydXN0IEphcGFuIENvLiwgTHRkLjEdMBsGA1UEAxMUU2VjdXJlU2lnbiBS
b290IENBMTQwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDF0nqh1oq/
FjHQmNE6lPxauG4iwWL3pwon71D2LrGeaBLwbCRjOfHw3xDG3rdSINVSW0KZnvOg
vlIfX8xnbacuUKLBl422+JX1sLrcneC+y9/3OPJH9aaakpUqYllQC6KxNedlsmGy
6pJxaeQp8E+BgQQ8sqVb1MWoWWd7VRxJq3qdwudzTe/NCcLEVxLbAQ4jeQkHO6Lo
/IrPj8BGJJw4J+CDnRugv3gVEOuGTgpa/d/aLIJ+7sr2KeH6caH3iGicnPCNvg9J
kdjqOvn90Ghx2+m1K06Ckm9mH+Dw3EzsytHqunQG+bOEkJTRX45zGRBdAuVwpcAQ
0BB8b8VYSbSwbprafZX1zNoCr7gsfXmPvkPx+SgojQlD+Ajda8iLLCSxjVIHvXib
y8posqTdDEx5YMaZ0ZPxMBoH064iwurO8YQJzOAUbn8/ftKChazcqRZOhaBgy/ac
18izju3Gm5h1DVXoX+WViwKkrkMpKBGk5hIwAUt1ax5mnXkvpXYvHUC0bcl9eQjs
0Wq2XSqypWa9a4X0dFbD9ed1Uigspf9mR6XU/v6eVL9lfgHWMI+lNpyiUBzuOIAB
SMbHdPTGrMNASRZhdCyvjG817XsYAFs2PJxQDcqSMxDxJklt33UkN4Ii1+iW/RVL
ApY+B3KVfqs9TC7XyvDf4Fg/LS8EmjijAQIDAQABo0IwQDAPBgNVHRMBAf8EBTAD
AQH/MA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUBpOjCl4oaTeqYR3r6/wtbyPk
86AwDQYJKoZIhvcNAQEMBQADggIBAJaAcgkGfpzMkwQWu6A6jZJOtxEaCnFxEM0E
rX+lRVAQZk5KQaID2RFPeje5S+LGjzJmdSX7684/AykmjbgWHfYfM25I5uj4V7Ib
ed87hwriZLoAymzvftAj63iP/2SbNDefNWWipAA9EiOWWF3KY4fGoweITedpdopT
zfFP7ELyk+OZpDc8h7hi2/DsHzc/N19DzFGdtfCXwreFamgLRB7lUe6TzktuhsHS
DCRZNhqfLJGP4xjblJUK7ZGqDpncllPjYYPGFrojutzdfhrGe0K22VoF3Jpf1d+4
2kd92jjbrDnVHmtsKheMYc2xbXIBw8MgAGJoFjHVdqqGuw6qnsb58Nn4DSEC5MUo
FlkRudlpcyqSeLiSV5sI8jrlL5WwWLdrIBRtFO8KvH7YVdiI2i/6GaX7i+B/OfVy
K4XELKzvGUWSTLNhB9xNH27SgRNcmvMSZ4PPmz+Ln52kuaiWA3rF7iDeM9ovnhp6
dB7h7sxaOgTdsxoEqBRjrLdHEoOabPXm6RUVkRqEGQ6UROcSjiVbgGcZ3GOTEAtl
Lor6CZpO2oYofaphNdgOpygau1LgePhsumywbrmHXumZNTfxPWQrqaA0k89jL9WB

365jJ6UeTo3cKXhZ+PmhIIynJkBugnLNeLLIjzwec+fBH7/PzqUqm9tEZDKgu39c
JRNItX+S
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICIzCCAamgAwIBAgIUFhXHw9hJp75pDIqI7fBw+d23PocwCgYIKoZIzj0EAwMw
UTELMAkGA1UEBhMCSlAxIzAhBgNVBAoTGkN5YmVydHJ1c3QgSmFwYW4gQ28uLCBM
dGQuMR0wGwYDVQQDExRTZWN1cmVTaWduIFJvb3QgQ0ExNTAeFw0yMDA0MDgwODMy
NTZaFw00NTA0MDgwODMyNTZaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKExpDeWJl
cnRydXN0IEphcGFuIENvLiwgTHRkLjEdMBsGA1UEAxMUU2VjdXJlU2lnbiBSb290
IENBMTUwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAAQLUHSNZDKZmbPSYAi4Io5GdCx4
wCtELW1fHcmuS1Iggz24FG1Th2CeX2yF2wYUleDHKP+dX+Sq8bOLbe1PL0vJSpSR
ZHX+AezB2Ot6lHhWGENfa4HL9rzatAy2KZMIaY+jQjBAMA8GA1UdEwEB/wQFMAMB
Af8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBTrQciu/NWeUUj1vYv0hyCTQSvT
9DAKBggqhkjOPQQDAwNoADBlAjEA2S6Jfl5OpBEHvVnCB96rMjhTKkZEBhd6zlHp
4P9mLQlO4E/0BdGF9jVg3PVys0Z9AjBEmEYagoUeYWmJSwdLZrWeqrqgHkHZAXQ6
bkU6iYAZezKYVWOr62Nuk22rGwlgMU4=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDuDCCAqCgAwIBAgIQDPCOXAgWpa1Cf/DrJxhZ0DANBgkqhkiG9w0BAQUFADBI
MQswCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3QgQ29ycG9yYXRpb24x
FzAVBgNVBAMTDlNlY3VyZVRydXN0IENBMB4XDTA2MTEwNzE5MzExOFoXDTI5MTIz
MTE5NDA1NVowSDELMAkGA1UEBhMCVVMxIDAeBgNVBAoTF1NlY3VyZVRydXN0IENv
cnBvcmF0aW9uMRcwFQYDVQQDEw5TZWN1cmVUcnVzdCBDQTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAKukgeWVzfX2FI7CT8rU4niVWJxB4Q2ZQCQXOZEz
Zum+4YOvYlyJ0fwkW2Gz4BERQRwdbvC4u/jep4G6pkjGnx29vo6pQT64lO0pGtSO
0gMdA+9tDWccV9cGrcrI9f4Or2YlSASWC12juhbDCE/RRvgUXPLIXgGZbf2IzIao
wW8xQmxSPmjL8xk037uHGFaAJsTQ3MBv396gwpEWoGQRS0S8Hvbn+mPeZqx2pHGj
7DaUaHp3pLHnDi+BeuK1cobvomuL8A/b01k/unK8RCSc43Oz969XL0Imnal0ugBS
8kvNU3xHCzaFDmapCJcWNFfBZveA4+1wVMeT4C4oFVmHursCAwEAAaOBnTCBmjAT
BgkrBgEEAYI3FAIEBh4EAEMAQTALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB
/zAdBgNVHQ4EFgQUQjK2FvoE/f5dS3rD/fdMQB1aQ68wNAYDVR0fBC0wKzApoCeg
JYYjaHR0cDovL2NybC5zZWN1cmV0cnVzdC5jb20vU1RDQS5jcmwwEAYJKwYBBAGC
NxUBBAMCAQAwDQYJKoZIhvcNAQEFBQADggEBADDtT0rhWDpSclu1pqNlGKa7UTt3
6Z3q059c4EVlew3KW+JwULKUBRSuSceNQQcSc5R+DCMh/bwQf2AQWnL1mA6s7Ll/
3XpvXdMc9P+IBWlCqQVxyLesJugutIxq/3HcuLHfmbx8IVQr5Fiiu1cprp6poxkm
D5kuCLDv/WnPmRoJjeOnnyvJNjR7JLN4TJUXpAYmHrZkUjZfYGfZnMUFdAvnZyPS
CPyI6a6Lf+Ew9Dd+/cYy2i2eRDAwbO4H3tI0/NL/QPZL9GZGBlSm8jIKYyYwa5vR
3ItHuuG51WLQoqD0ZwV4KWMabwTW+MZMo5qxN7SN5ShLHZ4swrhovO0C7jE=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDvDCCAqSgAwIBAgIQB1YipOjUiolN9BPI8PjqpTANBgkqhkiG9w0BAQUFADBK
MQswCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3QgQ29ycG9yYXRpb24x

GTAXBgNVBAMTEFNlY3VyZSBHbG9iYWwgQ0EwHhcNMDYxMTA3MTk0MjI4WhcNMjkx
MjMxMTk1MjA2WjBKMQswCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3Qg
Q29ycG9yYXRpb24xGTAXBgNVBAMTEFNlY3VyZSBHbG9iYWwgQ0EwggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQCvNS7YrGxVaQZx5RNoJLNP2MwhR/jxYDiJ
iQPpvepeRlMJ3Fz1Wuj3RSoC6zFh1ykzTM7HfAo3fg+6MpjhHZevj8fcyTiW89sa
/FHtaMbQbqR8JNGuQsiWUGMu4P51/pinX0kuleM5M2SOHqRfkNJnPLLZ/kG5VacJ
jnIFHovdRIWCQtBJwB1g8NEXLJXr9qXBkqPFwqcIYA1gBBCWeZ4WNOaptvolRTnI
HmX5k/Wq8VLcmZg9pYYaDDUz+kulBAYVHDGA76oYa8J719rO+TMg1fW9ajMtgQT7
sFzUnKPiXB3jqUJ1XnvUd+85VLrJChgbEplJL4hL/VBi0XPnj3pDAgMBAAGjgZ0w
gZowEwYJKwYBBAGCNxQCBAYeBABDAEEwCwYDVR0PBAQDAgGGMA8GA1UdEwEB/wQF
MAMBAf8wHQYDVR0OBBYEFK9EBMJBfkiD2045AuzshHrmzsmkMDQGA1UdHwQtMCsw
KaAnoCWGI2h0dHA6Ly9jcmwuc2VjdXJldHJ1c3QuY29tL1NHQ0EuY3JsMBAGCSsG
AQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEBBQUAA4IBAQBjGghAfaReUw132HquHw0L
URYD7xh8yOOvaliTFGCRsoTciE6+OYo68+aCiV0BN7OrJKQVDpI1WkpEXk5X+nXO
H0jOZvQ8QCaSmGwb7iRGDBezUqXbpZGRzzfTb+cnCDpOGR86p1hcF895P4vkp9Mm
I50mD1hp/Ed+stCNi5O/KU9DaXR2Z0vPB4zmAve14bRDtUstFJ/53CYNv6ZHdAbY
iNE6KTCEztI5gGIbqMdXSbxqVVFnFUq+NQfk1XWYN3kwFNspnWzFacxHVaIw98xc
f8LDmBxrThaA63p4ZUWiABqvDA1VZDRIuJK58bRQKfJPIx/abKwfROHdI3hRW8cW
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICODCCAb6gAwIBAgIJANZdm7N4gS7rMAoGCCqGSM49BAMDMGExCzAJBgNVBAYT
AkpQMSUwIwYDVQQKExxTRUNPTSBUcnVzdCBTeXN0ZW1zIENPLixMVEQuMSswKQYD
VQQDEyJTZWN1cml0eSBDb21tdW5pY2F0aW9uIEVDQyBSb290Q0ExMB4XDTE2MDYx
NjA1MTUyOFoXDTM4MDExODA1MTUyOFowYTELMAkGA1UEBhMCSlAxJTAjBgNVBAoT
HFNFQ09NIFRydXN0IFN5c3RlbXMgQ08uLExURC4xKzApBgNVBAMTIlNlY3VyaXR5
IENvbW11bmljYXRpb24gRUNDIFJvb3RDQTEwdjAQBgcqhkjOPQIBBgUrgQQAIgNi
AASkpW9gAwPDvTH00xecK4R1rOX9PVdu12O/5gSJko6BnOPpR27KkBLIE+Cnnfdl
dB9sELLo5OnvbYUymUSxXv3MdhDYW72ixvnWQuRXdtyQwjWpS4g8EkdtXP9JTxpK
ULGjQjBAMB0GA1UdDgQWBBSGHOf+LaVKiwj+KBH6vqNm+GBZLzAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkjOPQQDAwNoADBlAjAVXUI9/Lbu
9zuxNuie9sRGKEkz0FhDKmMpzE2xtHqiuQ04pV1IKv3LsnNdo4gIxwwCMQDAqy0O
be0YottT6SXbVQjgUMzfRGEWgqtJsLKB7HOHeLRMsmIbEvoWTSVLY70eN9k=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDdzCCAl+gAwIBAgIBADANBgkqhkiG9w0BAQsFADBdMQswCQYDVQQGEwJKUDEl
MCMGA1UEChMcU0VDT00gVHJ1c3QgU3lzdGVtcyBDTy4sTFRELjEnMCUGA1UECxMe
U2VjdXJpdHkgQ29tbXVuaWNhdGlvbiBSb290Q0EyMB4XDTA5MDUyOTA1MDAzOVoX
DTI5MDUyOTA1MDAzOVowXTELMAkGA1UEBhMCSlAxJTAjBgNVBAoTHFNFQ09NIFRy
dXN0IFN5c3RlbXMgQ08uLExURC4xJzAlBgNVBAsTHlNlY3VyaXR5IENvbW11bmlj
YXRpb24gUm9vdENBMjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANAV
OVKxUrO6xVmCxF1SrjpDZYBLx/KWvNs2l9amZIyoXvDjChz335c9S672XewhtUGr
zbl+dp+++T42NKA7wfYxEUV0kz1XgMX5iZnK5atq1LXaQZAQwdbWQonCv/Q4EpVM
VAX3NuRFg3sUZdbcDE3R3n4MqzvEFb46VqZab3ZpUql6ucjrappdUtAtCms1FgkQ
hNBqyjoGADdH5H5XTz+L62e4iKrFvlNVspHEfbmwhRkGeC7bYRr6hfVKkaHnFtWO

ojnflLhwHyg/i/xAXmODPIMqGplrz95Zajv8bxbXH/1KEOtOghY6rCcMU/Gt1SSw
awNQwS08Ft1ENCcadfsCAwEAAaNCMEAwHQYDVR0OBBYEFAqFqXdlBZh8QIH4D5cs
OPEK7DzPMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3
DQEBCwUAA4IBAQBMOqNErLlFsceTfsgLCkLfZOoc7llsCLqJX2rKSpWeeo8HxdpF
coJxDjrSzG+ntKEju/Ykn8sX/oymzsLS28yN/HH8AynBbF0zX2S2ZTuJbxh2ePXc
okgfGT+Ok+vx+hfuzU7jBBJV1uXk3fs+BXziHV7Gp7yXT2g69ekuCkO2r1dcYmh8
t/2jioSgrGK+KwmHNPBqAbubKVY8/gA3zyNs8U6qtnRGEmyR7jTV7JqR50S+kDFy
1UkC9gLl9B/rfNmWVan/7Ir5mUf/NVoCqgTLiluHcSmRvaS0eg29mvVXIwAHIRc/
SjnRBUkLp7Y3gaVdjKozXoEofKd9J+sAro03
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIID3TCCAsWgAwIBAgIBADANBgkqhkiG9w0BAQsFADCBjzELMAkGA1UEBhMCVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxJTAjBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xMjAwBgNVBAMTKVN0YXJmaWVs
ZCBSb290IENlcnRpZmljYXRlIEF1dGhvcml0eSAtIEcyMB4XDTA5MDkwMTAwMDAw
MFoXDTM3MTIzMTIzNTk1OVowgY8xCzAJBgNVBAYTAlVTMRAwDgYDVQQIEwdBcml6
b25hMRMwEQYDVQQHEwpTY290dHNkYWxlMSUwIwYDVQQKExxTdGFyZmllbGQgVGVj
aG5vbG9naWVzLCBJbmMuMTIwMAYDVQQDEylTdGFyZmllbGQgUm9vdCBDZXJ0aWZp
Y2F0ZSBBdXRob3JpdHkgLSBHMjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAL3twQP89o/8ArFvW59I2Z154qK3A2FWGMNHttfKPTUuiUP3oWmb3ooa/RMg
nLRJdzIpVv257IzdIvpy3Cdhl+72WoTsbhm5iSzchFvVdPtrX8WJpRBSiUZV9Lh1
HOZ/5FSuS/hVclcCGfgXcVnrHigHdMWdSL5stPSksPNkN3mSwOxGXn/hbVNMYq/N
Hwtjuzqd+/x5AJhhdM8mgkBj87JyahkNmcrUDnXMN/uLicFZ8WJ/X7NfZTD4p7dN
dloedl40wOiWVpmKs/B/pM293DIxfJHP4F8R+GuqSVzRmZTRouNjWwl2tVZi4Ut0
HZbUJtQIBFnQmA4O5t78w+wfkPECAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAO
BgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFHwMMh+n2TB/xH1oo2Kooc6rB1snMA0G
CSqGSIb3DQEBCwUAA4IBAQARWfolTwNvlJk7mh+ChTnUdgWUXuEok21iXQnCoKjU
sHU48TRqneSfioYmUeYs0cYtbpUgSpIB7LiKZ3sx4mcujJUDJi5DnUox9g61DLu3
4jd/IroAow57UvtruzvE03lRTs2Q9GcHGcg8RnoNAX3FWOdt5oUwF5okxBDgBPfg
8n/Uqgr/Qh037ZTlZFkSIHc40zI+OIF1lnP6aI+xy84fxez6nH7PfrHxBy22/L/K
pL/QlwVKvOoYKAKQvVR4CSFx09F9HdkWsKlhPdAKACL8x3vLCWRFCztAgfd9fDL1
mMpYjn0q7pBZc2T5NnReJaH1ZgUufzkVqSr7UIuOhWn0
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIID7zCCAtegAwIBAgIBADANBgkqhkiG9w0BAQsFADCBmDELMAkGA1UEBhMCVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxJTAjBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xOzA5BgNVBAMTMlN0YXJmaWVs
ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dGhvcml0eSAtIEcyMB4XDTA5
MDkwMTAwMDAwMFoXDTM3MTIzMTIzNTk1OVowgZgxCzAJBgNVBAYTAlVTMRAwDgYD
VQQIEwdBcml6b25hMRMwEQYDVQQHEwpTY290dHNkYWxlMSUwIwYDVQQKExxTdGFy
ZmllbGQgVGVjaG5vbG9naWVzLCBJbmMuMTswOQYDVQQDEzJTdGFyZmllbGQgU2Vy
dmljZXMgUm9vdCBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkgLSBHMjCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBANUMOsQq+U7i9b4Zl1+OiFOxHz/Lz58gE20p
OsgPfTz3a3Y4Y9k2YKibXlwAgLIvWX/2h/klQ4bnaRtSmpDhcePYLQ1Ob/bISdm2

8xpWriu2dBTrz/sm4xq6HZYuajtYlIlHVv8loJNwU4PahHQUw2eeBGg6345AWh1K
Ts9DkTvnVtYAcMtS7nt9rjrnvDH5RfbCYM8TWQIrgMw0R9+53pBlbQLPLJGmpufe
hRhJfGZOozptqbXuNC66DQO4M99H67FrjSXZm86B0UVGMpZwh94CDklDhbZsc7tk
6mFBrMnUVN+HL8cisibMn1lUaJ/8viovxFUcdUBgF4UCVTmLfwUCAwEAAaNCMEAw
DwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFJxfAN+q
AdcwKziIorhtSpzyEZGDMA0GCSqGSIb3DQEBCwUAA4IBAQBLNqaEd2ndOxmfZyMI
bw5hyf2E3F/YNoHN2BtBLZ9g3ccaaNnRbobhiCPPE95Dz+I0swSdHynVv/heyNXB
ve6SbzJ08pGCL72CQnqtKrcgfU28elUSwhXqvfdqlS5sdJ/PHLTyxQGjhdByPq1z
qwubdQxtRbeOlKyWN7Wg0I8VRw7j6IPdj/3vQQF3zCepYoUz8jcI73HPdwbeyBkd
iEDPfUYd/x7H4c7/I9vG+o1VTqkC50cRRj70/b17KSa7qWFiNyi2LSr2EIZkyXCn
0q23KXB56jzaYyWf/Wi3MOxw+3WKt21gZ7IeyLnp2KhvAotnDU0mV3HaIPzBSlCN
sSi6
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFujCCA6KgAwIBAgIJALtAHEP1Xk+wMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkNIMRUwEwYDVQQKEwxTd2lzc1NpZ24gQUcxHzAdBgNVBAMTFlN3aXNzU2ln
biBHb2xkIENBIC0gRzIwHhcNMDYxMDI1MDgzMDM1WhcNMzYxMDI1MDgzMDM1WjBF
MQswCQYDVQQGEwJDSDEVMBMGA1UEChMMU3dpc3NTaWduIEFHMR8wHQYDVQQDExZT
d2lzc1NpZ24gR29sZCBDQSAtIEcyMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIIC
CgKCAgEAr+TufoskDhJuqVAtFkQ7kpJcyrhdhJJCEyq8ZVeCQD5XJM1QiyUqt2/8
76LQwB8CJEoTlo8jE+YoWACjR8cGp4QjK7u9lit/VcyLwVcfDmJlD909Vopz2q5+
bbqBHH5CjCA12UNNhPqE21Is8w4ndwtrvxEvcnifLtg+5hg3Wipy+dpikJKVyh+c
6bM8K8vzARO/Ws/BtQpgvd21mWRTuKCWs2/iJneRjOBiEAKfNA+k1ZIzUd6+jbqE
emA8atufK+ze3gE/bk3lUIbLtK/tREDFylqM2tIrfKjuvqblCqoOpd8FUrdVxyJd
MmqXl2MT28nbeTZ7hTpKxVKJ+STnnXepgv9VHKVxaSvRAiTysybUa9oEVeXBCsdt
MDeQKuSeFDNeFhdVxVu1yzSJkvGdJo+hB9TGsnhQ2wwMC3wLjEHXuendjIj3o02y
MszYF9rNt85mndT9Xv+9lz4pded+p2JYryU0pUHHPbwNUMoDAw8IWh+Vc3hiv69y
FGkOpeUDDniOJihC8AcLYiAQZzlG+qkDzAQ4embvIIO1jEpWjpEA/I5cgt6IoMPi
aG59je883WX0XaxR7ySArqpWl2/5rX3aYT+YdzylkbYcjCbaZaIJbcHiVOO5ykxM
gI93e2CaHt+28kgeDrpOVG2Y4OGiGqJ3UM/EY5LsRxmd6+ZrzsECAwEAAaOBrDCB
qTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUWyV7
lqRlUX64OfPAeGZe6Drn8O4wHwYDVR0jBBgwFoAUWyV7lqRlUX64OfPAeGZe6Drn
8O4wRgYDVR0gBD8wPTA7BglghXQBWQECAQEwLjAsBggrBgEFBQcCARYgaHR0cDov
L3JlcG9zaXRvcnkuc3dpc3NzaWduLmNvbS8wDQYJKoZIhvcNAQEFBQADggIBACe6
45R88a7A3hfm5djV9VSwg/S7zV4Fe0+fdWavPOhWfvxyeDgD2StiGwC5+OlgzczO
UYrHUDFu4Up+GC9pWbY9ZIEr44OE5iKHjn3g7gKZYbge9LgriBIWhMIxkziWMaa5
O1M/wySTVltpkuzFwbs4AOPsF6m43Md8AYOfMke6UiI0HTJ6CVanfCU2qT1L2sCC
bwq7EsiHSycR+R4tx5M/nttfJmtS2S6K8RTGRI0Vqbe/vd6mGu6uLftIdxf+u+yv
GPUqUfA5hJeVbG4bwyvEdGB5JbAKJ9/fXtI5z0V9QkvfsywexcZdylU6oJxpmo/a
77KwPJ+HbBIrZXAVUjEaJM9vMSNQH4xPjyPDdEFjHFWoFN0+4FFQz/EbMFYOkrCC
hdiDyyJkvC24JdVUorgG6q2SpCSgwYa1ShNqR88uC1aVVMvOmttqtKay20EIhid3
92qgQmwLOM7XdVAyksLfKzAiSNDVQTglXaTpXZ/GlHXQRf0wl0OPkKsKx4ZzYEpp
Ld6leNcG2mqeSz53OiATIgHQv2ieY2BrNU0LbbqhPcCT4H8js1WtciVORvnSFu+w
ZMEBnunKoGqYDs/YYPIvSbjkQuE4NRb0yG5P94FW6LqjviOvrv1vA+ACOzB2+htt
Qc8Bsem4yWb02ybzOqR08kkkW8mw0FfB+j564ZfJ
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFkzCCA3ugAwIBAgIUQ/oMX04bgBhE79G0TzUfRPSA7cswDQYJKoZIhvcNAQEL
BQAwUTELMAkGA1UEBhMCQ0gxFTATBgNVBAoTDFN3aXNzU2lnbiBBRzErMCkGA1UE
AxMiU3dpc3NTaWduIFJTQSBUTFMgUm9vdCBDQSAyMDIyIC0gMTAeFw0yMjA2MDgx
MTA4MjJaFw00NzA2MDgxMTA4MjJaMFExCzAJBgNVBAYTAkNIMRUwEwYDVQQKEwxT
d2lzc1NpZ24gQUcxKzApBgNVBAMTIlN3aXNzU2lnbiBSU0EgVExTIFJvb3QgQ0Eg
MjAyMiAtIDEwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDLKmjiC8NX
vDVjvHClO/OMPE5Xlm7DTjak9gLKHqquuN6orx122ro10JFwB9+zBvKK8i5VUXu7
LCTLf5ImgKO0lPaCoaTo+nUdWfMHamFk4saMla+ju45vVs9xzF6BYQ1t8qsCLqSX
5XH8irCRIFucdFJtrhUnWXjyCcplDn/L9Ovn3KlMd/YrFgSVrpxxpT8q2kFC5zyE
EPThPYxr4iuRR1VPuFa+Rd4iUU1OKNlfGUEGjw5NBuBwQCMBauTLE5tzrE0USJIt
/m2n+IdreXXhvhCxqohAWVTXz8TQm0SzOGlkjIHRI36qOTw7D59Ke4LKa2/KIj4x
0LDQKhySio/YGZxH5D4MucLNvkEM+KRHBdvBFzA4OmnczcNpI/2aDwLOEGrOyvi5
KaM2iYauC8BPY7kGWUleDsFpswrzd34unYyzJ5jSmY0lpx+Gs6ZUcDj8fV3oT4MM
0ZPlEuRU2j7yrTrePjxF8CgPBrnh25d7mUWe3f6VWQQvdT/TromZhqwUtKiE+shd
OxtYk8EXlFXIC+OCeYSf8wCENO7cMdWP8vpPlkwGqnj73mSiI80fPsWMvDdUDrta
clXvyFu1cvh43zcgTFeRc5JzrBh3Q4IgaezprClG5QtO+DdziZaKHG29777YtvTK
wP1H8K4LWCDFyB02rpeNUIMmJCn3nTsPBQIDAQABo2MwYTAPBgNVHRMBAf8EBTAD
AQH/MA4GA1UdDwEB/wQEAwIBBjAfBgNVHSMEGDAWgBRvjmKLk0Ow4UD2p8P98Q+4
DxU4pTAdBgNVHQ4EFgQUb45ii5NDsOFA9qfD/fEPuA8VOKUwDQYJKoZIhvcNAQEL
BQADggIBAKwsKUF9+lz1GpUYvyypiqkkVHX1uECry6gkUSsYP2OprphWKwVDIqO3
10aewCoSPY6WlkDfDDOLazeROpW7OSltwAJsipQLBwJNGD77+3v1dj2b9l4wBlgz
Hqp41eZUBDqyggmNzhYzWUUo8aWjlw5DI/0LIICQ/+Mmz7hkkeUFjxOgdg3XNwwQ
iJb0Pr6VvfHDffCjw3lHC1ySFWPtUnWK50Zpy1FVCypM9fJkT6lc/2cyjlUtMoIc
gC9qkfjLvH4YoiaoLqNTKIftV+Vlek4ASltOU8liNr3CjlvrzG4ngRhZi0Rjn9UM
ZfQpZX+RLOV/fuiJz48gy20HQhFRJjKKLjpHE7iNvUcNCfAWpO2Whi4Z2L6MOuhF
LhG6rlrnub+xzI/goP+4s9GFe3lmozm1O2bYQL7Pt2eLSMkZJVX8vY3PXtpOpvJp
zv1/THfQwUY1mFwjmwJFQ5Ra3bxHrSL+ul4vkSkphnsh3m5kt8sNjzdbowhq6/Td
Ao9QAwKxuDdollDruF/UKIqlIgyKhPBZLtU30WHlQnNYKoH3dtvi4k0NX/a3vgW0
rk4N3hY9A4GzJl5LuEsAz/+MF7psYC0nhzck5npgL7XTgwSqT0N1osGDsieYK7EO
gLrAhV5Cud+xYJHT6xh+cHiudoO+cVrQkOPKwRYlZ0rwtnu64ZzZ
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDwzCCAqugAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgjELMAkGA1UEBhMCREUx
KzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnByaXNlIFNlcnZpY2VzIEdtYkgxHzAd
BgNVBAsMFlQtU3lzdGVtcyBUcnVzdCBDZW50ZXIxJTAjBgNVBAMMHFQtVGVsZVNl
YyBHbG9iYWxSb290IENsYXNzIDIwHhcNMDgxMDAxMTA0MDE0WhcNMzMxMDAxMjM1
OTU5WjCBgjELMAkGA1UEBhMCREUxKzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnBy
aXNlIFNlcnZpY2VzIEdtYkgxHzAdBgNVBAsMFlQtU3lzdGVtcyBUcnVzdCBDZW50
ZXIxJTAjBgNVBAMMHFQtVGVsZVNlYyBHbG9iYWxSb290IENsYXNzIDIwggEiMA0G
CSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCqX9obX+hzkeXaXPSi5kfl82hVYAUd
AqSzm1nzHoqvNK38DcLZSBnuaY/JIPwhqgcZ7bBcrGXHX+0CfHt8LRvWurmAwhiC
FoT6ZrAIxlQjgeTNuUk/9k9uN0goOA/FvudocP05l03Sx5iRUKrERLMjfTlH6VJi
1hKTXrcxlkIF+3anHqP1wvzpesVsqXFP6st4vGCvx9702cu+fjOlbpSD8DT6Iavq

jnKgP6TeMFvvhk1qlVtDRKgQFRzlAVfFmPHmBiiRqiDFt1MmUUOyCxGVWOHAD3bZ
wI18gfNycJ5v/hqO2V81xrJvNHy+SE/iWjnX2J14np+GPgNeGYtEotXHAgMBAAGj
QjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBS/
WSA2AHmgoCJrjNXyYdK4LMuCSjANBgkqhkiG9w0BAQsFAAOCAQEAMQOiYQsfdOhy
NsZt+U2e+iKo4YFWz827n+qrkRk4r6p8FU3ztqONpfSO9kSpp+ghla0+AGIWiPAC
uvxhI+YzmzB6azZie60EI4RYZeLbK4rnJVM3YlNfvNoBYimipidx5joifsFvHZVw
IEoHNN/q/xWA5brXethbdXwFeilHfkCoMRN3zUA7tFFHei4R40cR3p1m0IvVVGb6
g1XqfMIpiRvpb7PO4gWEyS8+eIVibslfwXhjdFjASBgMmTnrpMwatXlajRWc2BQN
9noHV8cigwUtPJslJj0Ys6lDfMjIq2SPDqO/nBudMNva0BkuqjzxzOAduTNrRlP
BSeOE6Fuwg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDwzCCAqugAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgjELMAkGA1UEBhMCREUx
KzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnByaXNlIFNlcnZpY2VzIEdtYkgxHzAd
BgNVBAsMFlQtU3lzdGVtcyBUcnVzdCBDZW50ZXIxJTAjBgNVBAMMHFQtVGVsZVNl
YyBHbG9iYWxSb290IENsYXNzIDMwHhcNMDgxMDAxMTAyOTU2WhcNMzMxMDAxMjM1
OTU5WjCBgjELMAkGA1UEBhMCREUxKzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnBy
aXNlIFNlcnZpY2VzIEdtYkgxHzAdBgNVBAsMFlQtU3lzdGVtcyBUcnVzdCBDZW50
ZXIxJTAjBgNVBAMMHFQtVGVsZVNlYyBHbG9iYWxSb290IENsYXNzIDMwggEiMA0G
CSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC9dZPwYiJvJK7genasfb3ZJNW4t/zN
8ELg63iIVl6bmlQdTQyK9tPPcPRStdiTBONGhnFBSivwKixVA9ZIw+A5OO3yXDw/
RLyTPWGrTs0NvvAgJ1gORH8EGoel15YUNpDQSXuhdfsaa3Ox+M6pCSzyU9XDFES4
hqX2iys52qMzVNn6chr3IhUciJFrf2blw2qAsCTz34ZFiP0Zf3WHHx+xGwpzJFu5
ZeAsVMhg02YXP+HMVDNzkQI6pn97djmiH5a2OK61yJN0HZ65tOVgnS9W0eDrXltM
EnAMbEQgqxHY9Bn20pxSN+f6tsIxO0rUFJmtxxr1XV/6B7h8DR/Wgx6zAgMBAAGj
QjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBS1
A/d2O2GCahKqGFPrAyGUv/7OyjANBgkqhkiG9w0BAQsFAAOCAQEAVj3vlNW92nOy
WL6ukK2YJ5f+AbGwUgC4TeQbIXQbfsDuXmkqJa9c1h3a0nnJ85cp4IaH3gRZD/FZ
1GSFS5mvJQQeyUapl96Cshtwn5z2r3Ex3XsFpSzTucpH9sry9uetuUg/vBa3wW30
6gmv7PO15wWeph6KU1HWk4HMdJP2udqmJQV0eVp+QD6CSyYRMG7hP0HHRwA11fXT
91Q+gT3aSWqas+8QPebrb9HIIkfLzM8BMZLZGOMivgkeGj5asuRrDFR6fUNOuIml
e9eiPZaGzPImNC1qkp2aGtAw4l1OBLBfiyB+d8E9lYLRRpo7PHi4b6HQDWSieB4p
TpPDpFQUWw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIEYzCCA0ugAwIBAgIBATANBgkqhkiG9w0BAQsFADCB0jELMAkGA1UEBhMCVFIx
GDAWBgNVBAcTD0dlYnplIC0gS29jYWVsaTFCMEAGA1UEChM5VHVya2l5ZSBCaWxp
bXNlbCB2ZSBUZWtub2xvamlrIEFyYXN0aXJtYSBLdXJ1bXUgLSBUVUJJVEFLMS0w
KwYDVQQLEyRLYW11IFNlcnRpZmlrYXN5b24gTWVya2V6aSAtIEthbXUgU00xNjA0
BgNVBAMTLVRVQklUQUsgS2FtdSBTTSBU0wgS29rIFNlcnRpZmlrYXNpIC0gU3Vy
dW0gMTAeFw0xMzExMjUwODI1NTVaFw00MzEwMjUwODI1NTVaMIHSMQswCQYDVQQG
EwJUUjEYMBYGA1UEBxMPR2ViemUgLSBLb2NhZWxpMUIwQAYDVQQKEzlUdXJraXll
IEJpbGltc2VsIHZlIFRla25vbG9qaWsgQXJhc3Rpcm1hIEt1cnVtdSAtIFRVQklU
QUsxLTArBgNVBAsTJEthbXUgU2VydGlmaWthc3lvbiBNZXJrZXppIC0gS2FtdSBT

TTE2MDQGA1UEAxMtVFVCSVRBSyBLYW11IFNNIFNTTCBLb2sgU2VydGlmaWthc2kg
LSBTdXJ1bSAxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAr3UwM6q7
a9OZLBI3hNmNe5eA027n/5tQlT6QlVZC1xl8JoSNkvoBHToP4mQ4t4y86Ij5iySr
LqP1N+RAjhgleYN1Hzv/bKjFxlb4tO2KRKOrbEz8HdDc72i9z+SqzvBV96I01INr
N3wcwv61A+xXzry0tcXtAA9TNypN9E8Mg/uGz8v+jE69h/mniyFXnHrfA2eJLJ2X
YacQuFWQfw4tJzh03+f92k4S400VIgLI4OD8D62K18lUUMw7D8oWgITQUVbDjlZ/
iSIzL+aFCr2lqBs23tPcLG07xxO9WSMs5uWk99gL7eqQQESolbuT1dCANLZGeA4f
AJNG4e7p+exPFwIDAQABo0IwQDAdBgNVHQ4EFgQUZT/HiobGPN08VFw1+DrtUgxH
V8gwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEL
BQADggEBACo/4fEyjq7hmFxLXs9rHmoJ0iKpEsdeV31zVmSAhHqT5Am5EM2fKifh
AHe+SMg1qIGf5LgsyX8OsNJLN13qudULXjS99HMpw+0mFZx+CFOKWI3QSyjfwbPf
IPP54+M638yclNhOT8NrF7f3cuitZjO1JVOr4PhMqZ398g26rrnZqsZr+ZO7rqu4
lzwDGrpDxpa5RXI4s6ehlj2Re37AIVNMh+3yC1SVUZPVIqUNivGTDj5UDrDYyU7c
8jEyVupk+eq1nRZmQnLzf9OxMUP8pI4X8W0jq5Rm+K37DwhuJi1/FwcJsoz7UMCf
lo3Ptv0AnVoUmr8CRPXBwp8iXqIPoeM=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFjTCCA3WgAwIBAgIQQAE0jMIAAAAAAAAAATzyxjANBgkqhkiG9w0BAQwFADBQ
MQswCQYDVQQGEwJUVzESMBAGA1UEChMJVEFJV0FOLUNBMRAwDgYDVQQLEwdSb290
IENBMRswGQYDVQQDExJUV0NBIENZQkVSIFJvb3QgQ0EwHhcNMjIxMTIyMDY1NDI5
WhcNNDcxMTIyMTU1OTU5WjBQMQswCQYDVQQGEwJUVzESMBAGA1UEChMJVEFJV0FO
LUNBMRAwDgYDVQQLEwdSb290IENBMRswGQYDVQQDExJUV0NBIENZQkVSIFJvb3Qg
Q0EwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDG+Moe2Qkgfh1sTs6P
40czRJzHyWmqOlt47nDSkvgEs1JSHWdyKKHfi12VCv7qze33Kc7wb3+szT3vsxxF
avcokPFhV8UMxKNQXd7UtcsZyoC5dc4pztKFIuwCY8xEMCDa6pFbVuYdHNWdZsc/
34bKS1PE2Y2yHer43CdTo0fhYcx9tbD47nORxc5zb87uEB8aBs/pJ2DFTxnk684i
JkXXYJndzk834H/nY62wuFm40AZoNWDTNq5xQwTxaWV4fPMf88oon1oglWa0zbfu
j3ikRRjpJi+NmykosaS3Om251Bw4ckVYsV7r8Cibt4LK/c/WMw+f+5eesRycnupf
Xtuq3VTpMCEobY5583WSjCb+3MX2w7DfRFlDo7YDKPYIMKoNM+HvnKkHIuNZW0CP
2oi3aQiotyMuRAlZN1vH4xfyIutuOVLF3lSnmMlLIJXcRolftBL5hSmO68gnFSDA
S9TMfAxsNAwmmyYxpjyn9tnQS6Jk/zuZQXLB4HCX8SS7K8R0IrGsayIyJNN4KsDA
oS/xUgXJP+92ZuJF2A09rZXIx4kmyA+upwMu+8Ff+iDhcK2wZSA3M2Cw1a/XDBzC
kHDXShi8fgGwsOsVHkQGzaRP6AzRwyAQ4VRlnrZR0Bp2a0JaWHY06rc3Ga4udfmW
5cFZ95RXKSWNOkyrTZpB0F8mAwIDAQABo2MwYTAOBgNVHQ8BAf8EBAMCAQYwDwYD
VR0TAQH/BAUwAwEB/zAfBgNVHSMEGDAWgBSdhWEUfMFib5do5E83QOGt4A1WNzAd
BgNVHQ4EFgQUnYVhFHzBYm+XaORPN0DhreANVjcwDQYJKoZIhvcNAQEMBQADggIB
AGSPesRiDrWIzLjHhg6hShbNcAu3p4ULs3a2D6f/CIsLJc+o1IN1KriWiLb73y0t
tGlTITVX1olNc79pj3CjYcya2x6a4CD4bLubIp1dhDGaLIrdaqHXKGnK/nZVekZn
68xDiBaiA9a5F/gZbG0jAn/xX9AKKSM70aoK7akXJlQKTcKlTfjF/biBzysseKNn
TKkHmvPfXvt89YnNdJdhEGoHK4Fa0o635yDRIG4kqIQnoVesqlVYL9zZyvpoBJ7t
RCT5dEA7IzOrg1oYJkK2bVS1FmAwbLGg+LhBoF1JSdJlBTrq/p1hvIbZv97Tujqx
f36SNI7JAG7cmL3c7IAFrQI932XtCwP39xaEBDG6k5TY8hL4iuO/Qq+n1M0RFxbI
Qh0UqEL20kCGoE8jypZFVmAGzbdVAaYBlGX+bgUJurSkquLvWL69J1bY73NxW0Qz
8ppy6rBePm6pUlvscG21h483XjyMnM7k8M4MZ0HMzvaAq07MTFb1wWFZk7Q+ptq4
NxKfKjLji7gh7MMrZQzvIt6IKTtM1/r+t+FHvpw+PoP7UV31aPcuIYXcv/Fa4nzX
xeSDwWrruoBa3lwtcHb4yOWHh8qgnaHlIhInD0Q9HWzq1MKLL295q39QpsQZp6F6

t5b5wR9iWqJDB0BeJsas7a5wFsWqynKKTbDPAYsDP27X
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFQTCCAymgAwIBAgICDL4wDQYJKoZIhvcNAQELBQAwUTELMAkGA1UEBhMCVFcx
EjAQBgNVBAoTCVRBSVdBTi1DQTEQMA4GA1UECxMHUm9vdCBDQTEcMBoGA1UEAxMT
VFdDQSBHbG9iYWwgUm9vdCBDQTAeFw0xMjA2MjcwNjI4MzNaFw0zMDEyMzExNTU5
NTlaMFExCzAJBgNVBAYTAlRXMRIwEAYDVQQKEwlUQUlXQU4tQ0ExEDAOBgNVBAsT
B1Jvb3QgQ0ExHDAaBgNVBAMTE1RXQ0EgR2xvYmFsIFJvb3QgQ0EwggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCwBdvI64zEbooh745NnHEKH1Jw7W2CnJfF
10xORUnLQEK1EjRsGcJ0pDFfhQKX7EMzClPSnIyOt7h52yvVavKOZsTuKwEHktSz
0ALfUPZVr2YOy+BHYC8rMjk1Ujoog/h7FsYYuGLWRyWRzvAZEk2tY/XTP3VfKfCh
MBwqoJimFb3u/Rk28OKRQ4/6ytYQJ0lM793B8YVwm8rqqFpD/G2Gb3PpN0Wp8DbH
zIh1HrtsBv+baz4X7GGqcXzGHaL3SekVtTzWoWH1EfcFbx39Eb7QMAfCKbAJTibc
46KokWofwpFFiFzlmLhxpRUZyXx1EcxwdE8tmx2RRP1WKKD+u4ZqyPpcC1jcxkt2
yKsi2XMPpfRaAok/T54igu6idFMqPVMnaR1sjjIsZAAmY2E2TqNGtz99sy2sbZCi
laLOz9qC5wc0GZbpuCGqKX6mOL6OKUohZnkfs8O1CWfe1tQHRvMq2uYiN2DLgbYP
oA/pyJV/v1WRBXrPPRXAb94JlAGD1zQbzECl8LibZ9WYkTunhHiVJqRaCPgrdLQA
BDzfuBSO6N+pjWxnkjMdwLfS7JLIvgm/LCkFbwJrnu+8vyq8W8BQj0FwcYeyTbcE
qYSjMq+u7msXi7Kx/mzhkIyIqJdIzshNy/MGz19qCkKxHh53L46g5pIOBvwFItIm
4TFRfTLcDwIDAQABoyMwITAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB
/zANBgkqhkiG9w0BAQsFAAOCAgEAXzSBdu+WHdXltdkCY4QWwa6gcFGn90xHNcgL
1yg9iXHZqjNB6hQbbCEAwGxCGX6faVsgQt+i0trEfJdLjbDorMjupWkEmQqSpqsn
LhpNgb+E1HAerUf+/UqdM+DyucRFCCEK2mlpc3INvjT+lIutwx4116KD7+U4x6WF
H6vPNOw/KP4M8VeGTslV9xzU2KV9Bnpv1d8Q34FOIWWxtuEXeZVFBs5fzNxGiWNo
RI2T9GRwoD2dKAXDOXC4Ynsg/eTb6QihuJ49CcdP+yz4k3ZB3lLg4VfSnQO8d57+
nile98FRYB/e2guyLXW3Q0iT5/Z5xoRdgFlglPx4mI88k1HtQJAH32RjJMtOcQWh
15QaiDLxInQirqWm2BJpTGCjAu4r7NRjkgtevi92a6O2JryPA9gK8kxkRr05YuWW
6zRjESjMlfGt7+/cgFhI6Uu46mWs6fyAtbXIRfmswZ/ZuepiiI7E8UuDEq3mi4TW
nsLrgxifarsbJGAzcMzs9zLzXNl5fe+epP7JI8Mk7hWSsT2RTyaGvWZzJBPqpK5j
wa19hAM8EHiGG3njxPPyBJUgriOCxLM6AGK/5jYk4Ve6xx6QddVfP5VhK8E7zeWz
aGHQRiapIVJpLesux+t3zqY6tQMzT3bR51xUAV3LePTJDL/PEo4XLSNolOer/qmy
KwbQBM0=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDezCCAmOgAwIBAgIBATANBgkqhkiG9w0BAQUFADBfMQswCQYDVQQGEwJUVzES
MBAGA1UECgwJVEFJV0FOLUNBMRAwDgYDVQQLDAdSb290IENBMSowKAYDVQQDDCFU
V0NBIFJvb3QgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMDgwODI4MDcyNDMz
WhcNMzAxMjMxMTU1OTU5WjBfMQswCQYDVQQGEwJUVzESMBAGA1UECgwJVEFJV0FO
LUNBMRAwDgYDVQQLDAdSb290IENBMSowKAYDVQQDDCFUV0NBIFJvb3QgQ2VydGlm
aWNhdGlvbiBBdXRob3JpdHkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQCwfnK4pAOU5qfeCTiRShFAh6d8WWQUe7UREN3+v9XAu1bihSX0NXIP+FPQQeFE
AcK0HMMxQhZHhTMidrIKbw/lJVBPhYa+v5guEGcevhEFhgWQxFnQfHgQsIBct+HH
K3XLfJ+utdGdIzdjp9xCoi2SBBtQwXu4PhvJVgSLL1KbralW6cH/ralYhzC2gfeX
RfwZVzsrb+RH9JlF/h3x+JejiB03HFyP4HYlmlD4oFT/RJB2I9IyxsOrBr/8+7/z

rX2SYgJbKdM1o5OaQ2RgXbL6Mv87BK9NQGr5x+PvI/1ry+UPizgN7gr8/g+YnzAx
3WxSZfmLgb4i4RxYA7qRG4kHAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBBRqOFsmjd6LWvJPelSDGRjjCDWmujANBgkq
hkiG9w0BAQUFAAOCAQEAPNV3PdrfibqHDAhUaiBQkr6wQT25JmSDCi/oQMCXKCeC
MErJk/9q56YAf4lCmtYR5VPOL8zy2gXE/uJQxDqGfczafhAJO5I1KlOy/usrBdls
XebQ79NqZp4VKIV66IIArB6nCWlWQtNoURi+VJq/REG6Sb4gumlc7rh3zc5sH62D
lhh9DrUUOYTxKOkto557HnpyWoOzeW/vtPzQCqVYT0bf+215WfKEIlKuD8z7fDvn
aspHYcN6+NOSBB+4IIThNlQWx0DeO4pz3N/GCUzf7Nr/1FNCocnyYh0igzyXxfkZ
YiesZSLX0zzG5Y6yU8xJzrww/nsOM5D77dIUkR8Hrw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICQjCCAcmgAwIBAgIQNjqWjMlcsljN0AFdxeVXADAKBggqhkjOPQQDAzBjMQsw
CQYDVQQGEwJERTEnMCUGA1UECgweRGV1dHNjaGUgVGVsZWtvbSBTZWN1cml0eSBH
bWJIMSswKQYDVQQDDCJUZWxla29tIFNlY3VyaXR5IFRMUyBFQ0MgUm9vdCAyMDIw
MB4XDTIwMDgyNTA3NDgyMFoXDTQ1MDgyNTIzNTk1OVowYzELMAkGA1UEBhMCREUx
JzAlBgNVBAoMHkRldXRzY2hlIFRlbGVrb20gU2VjdXJpdHkgR21iSDErMCkGA1UE
AwwiVGVsZWtvbSBTZWN1cml0eSBUTFMgRUNDIFJvb3QgMjAyMDB2MBAGByqGSM49
AgEGBSuBBAAiA2IABM6//leov9Wq9xCazbzREaK9Z0LMkOsVGJDZos0MKiXrPk/O
tdKPD/M12kOLAoC+b1EkHQ9rK8qfwm9QMuU3ILYg/4gND21Ju9sGpIeQkpT0CdDP
f8iAC8GXs7s1J8nCG6NCMEAwHQYDVR0OBBYEFONyzG6VmUex5rNhTNHLq+O6zd6f
MA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMAoGCCqGSM49BAMDA2cA
MGQCMHVSi7ekEE+uShCLsoRbQuHmKjYC2qBuGT8lv9pZMo7k+5Dck2TOrbRBR2Di
z6fLHgIwN0GMZt9Ba9aDAEH9L1r3ULRn0SyocddDypwnJJGDSA3PzfdUga/sf+Rn
27iQ7t0l
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFszCCA5ugAwIBAgIQIZxULej27HF3+k7ow3BXlzANBgkqhkiG9w0BAQwFADBj
MQswCQYDVQQGEwJERTEnMCUGA1UECgweRGV1dHNjaGUgVGVsZWtvbSBTZWN1cml0
eSBHbWJIMSswKQYDVQQDDCJUZWxla29tIFNlY3VyaXR5IFRMUyBSU0EgUm9vdCAy
MDIzMB4XDTIzMDMyODEyMTY0NVoXDTQ4MDMyNzIzNTk1OVowYzELMAkGA1UEBhMC
REUxJzAlBgNVBAoMHkRldXRzY2hlIFRlbGVrb20gU2VjdXJpdHkgR21iSDErMCkG
A1UEAwwiVGVsZWtvbSBTZWN1cml0eSBUTFMgUlNBIFJvb3QgMjAyMzCCAiIwDQYJ
KoZIhvcNAQEBBQADggIPADCCAgoCggIBAO01oYGA88tKaVvC+1GDrib94W7zgRJ9
cUD/h3VCKSHtgVIs3xLBGYSJwb3FKNXVS2xE1kzbB5ZKVXrKNoIENqil/Cf2SfHV
cp6R+SPWcHu79ZvB7JPPGeplfohwoHP89v+1VmLhc2o0mD6CuKyVU/QBoCcHcqMA
U6DksquDOFczJZSfvkgdmOGjup5czQRxUX11eKvzWarE4GC+j4NSuHUaQTXtvPM6
Y+mpFEXX5lLRbtLevOP1Czvm4MS9Q2QTps70mDdsipWol8hHD/BeEIvnHRz+sTug
BTNoBUGCwQMrAcjnj02r6LX2zWtEtefdi+zqJbQAIldNsLGyMcEWzv/9FIS3R/qy
8XDe24tsNlikfLMR0cN3f1+2JeANxdKz+bi4d9s3cXFH42AYTyS2dTd4uaNir73J
co4vzLuu2+QVUhkHM/tqty1LkCiCc/4YizWN26cEar7qwU02OxY2kTLvtkCJkUPg
8qKrBC7m8kwOFjQgrIfBLX7JZkcXFBGk8/ehJImr2BrIoVyxo/eMbcgByU/J7MT8
rFEz0ciD0cmfHdRHNCk+y7AO+oMLKFjlKdw/fKifybYKu6boRhYPluV75Gp6SG12
mAWl3G0eQh5C2hrgUve1g8Aae3g1LDj1H/1Joy7SWWO/gLCMk3PLNaaZlSJhZQNg
+y+TS/qanIA7AgMBAAGjYzBhMA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUtqeX

gj10hZv3PJ+TmpV5dVKMbUcwDwYDVR0TAQH/BAUwAwEB/zAfBgNVHSMEGDAWgBS2
p5eCPXSFm/c8n5OalXl1UoxtRzANBgkqhkiG9w0BAQwFAAOCAgEAqMxhpr51nhVQ
pGv7qHBFfLp+sVr8WyP6Cnf4mHGCDG3gXkaqk/QeoMPhk9tLrbKmXauw1GLLXrtm
9S3ul0A8Yute1hTWjOKWi0FpkzXmuZlrYrShF2Y0pmtjxrlO8iLpWA1WQdH6DErw
M807u20hOq6OcrXDSvvpfeWxm4bu4uB9tPcy/SKE8YXJN3nptT+/XOR0so8RYgDd
GGah2XsjX/GO1WfoVNpbOms2b/mBsTNHM3dA+VKq3dSDz4V4mZqTuXNnQkYRIer+
CqkbGmVps4+uFrb2S1ayLfmlyOw7YqPta9BO1UAJpB+Y1zqlklkg5LB9zVtzaL1t
xKITDmcZuI1CfmwMmm6gJC3VRRvcxAIU/oVbZZfKTpBQCHpCNfnqwmbU+AGuHrS+
w6jv/naaoqYfRvaE7fzbzsQCzndILIyy7MMAo+wsVRjBfhnu4S/yrYObnqsZ38aK
L4x35bcF7DvB7L6Gs4a8wPfc5+pbrrLMtTWGS9DiP7bY+A4A7l3j941Y/8+LN+lj
X273CXE2whJdV/LItM3z7gLfEdxquVeEHVlNjM7IDiPCtyaaEBRx/pOyiriA8A4Q
ntOoUAw3gi/q4Iqd4Sw5/7W0cwDk90imc6y/st53BIe0o82bNSQ3+pCTE4FCxpgm
dTdmQRCsu/WU48IxK63nI1bMNSWSs1A=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFODCCAyCgAwIBAgIRAJW+FqD3LkbxezmCcvqLzZYwDQYJKoZIhvcNAQEFBQAw
NzEUMBIGA1UECgwLVGVsaWFTb25lcmExHzAdBgNVBAMMFlRlbGlhU29uZXJhIFJv
b3QgQ0EgdjEwHhcNMDcxMDE4MTIwMDUwWhcNMzIxMDE4MTIwMDUwWjA3MRQwEgYD
VQQKDAtUZWxpYVNvbmVyYTEfMB0GA1UEAwwWVGVsaWFTb25lcmEgUm9vdCBDQSB2
MTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAMK+6yfwIaPzaSZVfp3F
VRaRXP3vIb9TgHot0pGMYzHw7CTww6XScnwQbfQ3t+XmfHnqjLWCi65ItqwA3GV1
7CpNX8GH9SBlK4GoRz6JI5UwFpB/6FcHSOcZrr9FZ7E3GwYq/t75rH2D+1665I+X
Z75Ljo1kB1c4VWk0Nj0TSO9P4tNmHqTPGrdeNjPUtAa9GAH9d4RQAEX1jF3oI7x+
/jXh7VB7qTCNGdMJjmhnXb88lxhTuylixcpecsHHltTbLaC0H2kD7OriUPEMPPCs
81Mt8Bz17Ww5OXOAFshSsCPN4D7c3TxHoLs1iuKYaIu+5b9y7tL6pe0S7fyYGKkm
dtwoSxAgHNN/Fnct7W+A90m7UwW7XWjH1Mh1Fj+JWov3F0fUTPHSiXk+TT2YqGHe
Oh7S+F4D4MHJHIzTjU3TlTazN19jY5szFPAtJmtTfImMMsJu7D0hADnJoWjiUIMu
sDor8zagrC/kb2HCUQk5PotTubtn2txTuXZZNp1D5SDgPTJghSJRt8czu90VL6R4
pgd7gUY2BIbdeTXHlSw7sKMXNeVzH7RcWe/a6hBle3rQf5+ztCo3O3CLm1u5K7fs
slESl1MpWtTwEhDcTwK7EpIvYtQ/aUN8Ddb8WHUBiJ1YFkveupD/RwGJBmr2X7KQ
arMCpgKIv7NHfirZ1fpoeDVNAgMBAAGjPzA9MA8GA1UdEwEB/wQFMAMBAf8wCwYD
VR0PBAQDAgEGMB0GA1UdDgQWBBTwj1k4ALP1j5qWDNXr+nuqF+gTEjANBgkqhkiG
9w0BAQUFAAOCAgEAvuRcYk4k9AwI//DTDGjkk0kiP0Qnb7tt3oNmzqjMDfz1mgbl
dxSR651Be5kqhOX//CHBXfDkH1e3damhXwIm/9fH907eT/j3HEbAek9ALCI18Bmx
0GtnLLCo4MBANzX2hFxc469CeP6nyQ1Q6g2EdvZR74NTxnr/DlZJLo961gzmJ1Tj
TQpgcmLNkQfWpb/ImWvtxBnmq0wROMVvMeJuScg/doAmAyYp4Db29iBT4xdwNBed
Y2gea+zDTYa4EzAvXUYNR0PVG6pZDrlcjQZIrXSHX8f8MVRBE+LHIQ6e4B4N4cB7
Q4WQxYpYxmUKeFfyxiMPAdkgS94P+5KFdSpcc41teyWRyu5FrgZLAMzTsVlQ2jqI
OylDRl6XK1TOU2+NSueW+r9xDkKLfP0ooNBIytrEgUy7onOTJsjrDNYmiLbAJM+7
vVvrdX3pCI6GMyx5dwlppYn8s3CQh3aP0yK7Qs69cwsgJirQmz1wHiRszYd2qReW
t88NkvuOGKmYSdGe/mBEciG5Ge3C9THxOUiIkCR1VBatzvT4aRRkOfujuLpwQMcn
HL/EVlP6Y2XQ8xwOFvVrhlhNGNTkDY6lnVuR3HYkUD/GKvvZt5y11ubQ2egZixVx
SK236thZiNSQvxaz2emsWWFUyBy6ysHK4bkgTI86k4mloMy/0/Z1pHWWbVY=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFdDCCA1ygAwIBAgIPAWdfJ9b+euPkrL4JWwWeMA0GCSqGSIb3DQEBCwUAMEQx
CzAJBgNVBAYTAkZJMRowGAYDVQQKDBFUZWxpYSBGaW5sYW5kIE95ajEZMBcGA1UE
AwwQVGVsaWEgUm9vdCBDQSB2MjAeFw0xODExMjkxMTU1NTRaFw00MzExMjkxMTU1
NTRaMEQxCzAJBgNVBAYTAkZJMRowGAYDVQQKDBFUZWxpYSBGaW5sYW5kIE95ajEZ
MBcGA1UEAwwQVGVsaWEgUm9vdCBDQSB2MjCCAiIwDQYJKoZIhvcNAQEBBQADggIP
ADCCAgoCggIBALLQPwe84nvQa5n44ndp586dpAO8gm2h/oFlH0wnrI4AuhZ76zBq
AMCzdGh+sq/H1WKzej9Qyow2RCRj0jbpDIX2Q3bVTKFgcmfiKDOlyzG4OiIjNLh9
vVYiQJ3q9HsDrWj8soFPmNB06o3lfc1jw6P23pLCWBnglrvFxKk9pXSW/q/5iaq9
lRdU2HhE8Qx3FZLgmEKnpNaqIJLNwaCzlrI6hEKNfdWV5Nbb6WLEWLN5xYzTNTOD
n3WhUidhOPFZPY5Q4L15POdslv5e2QJltI5c0BE0312/UqeBAMN/mUWZFdUXyApT
7GPzmX3MaRKGwhfwAZ6/hLzRUssbkmbOpFPlob/E2wnW5olWK8jjfN7j/4nlNW4o
6GwLI1GpJQXrSPjdscr6bAhR77cYbETKJuFzxokGgeWKrLDiKca5JLNrRBH0pUPC
TEPlcDaMtjNXepUugqD0XBCzYYP2AgWGLnwtbNwDRm41k9V6lS/eINhbfpSQBGq6
WT0EBXWdN6IOLj3rwaRSg/7Qa9RmjtzG6RJOHSpXqhC8fF6CfaamyfItufUXJ63R
DolUK5X6wK0dmBR4M0KGCqlztft0DbcbMBnEWg4cJ7faGND/isgFuvGqHKI3t+ZI
pEYslOqodmJHixBTB0hXbOKSTbauBcvcwUpej6w9GU7C7WB1K9vBykLVAgMBAAGj
YzBhMB8GA1UdIwQYMBaAFHKs5DN5qkWH9v2sHZ7Wxy+G2CQ5MB0GA1UdDgQWBBRy
rOQzeapFh/b9rB2e1scvhtgkOTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUw
AwEB/zANBgkqhkiG9w0BAQsFAAOCAgEAoDtZpwmUPjaE0n4vOaWWl/oRrfxn83EJ
8rKJhGdEr7nv7ZbsnGTbMjBvZ5qsfl+yqwE2foH65IRe0qw24GtixX1LDoJt0nZi
0f6X+J8wfBj5tFJ3gh1229MdqfDBmgC9bXXYfef6xzijnHDoRnkDry5023X4blMM
A8iZGok1GTzTyVR8qPAs5m4HeW9q4ebqkYJpCh3DflminmtGFZhb069GHWLIzoBS
SRE/yQQSwxN8PzuKlts8oB4KtItUsiRnDe+Cy748fdHif64W1lZYudogsYMVoe+K
TTJvQS8TUoKU1xrBeKJR3Stwbbca+few4GeXVtt8YVMJAygCQMez2P2ccGrGKMOF
6eLtGpOg3kuYooQ+BXcBlj37tCAPnHICehIv1aO6UXivKitEZU61/Qrowc15h2Er
3oBXRb9n8ZuRXqWk7FlIEA04x7D6w0RtBPV4UBySllva9bguulvP5fBqnUsvWHMt
Ty3EHD70sz+rFQ47GUGKpMFXEmZxTPpT41frYpUJnlTd0cI8Vzy9OK2YZLe4A5pT
VmBds9hCG1xLEooc6+t9xnppxyd/pPiL8uSUZodL6ZQHCRJ5irLrdATczvREWeAW
ysUsWNc8e89ihmpQfTU2Zqf7N+cox9jQraVplI/owd8k+BsHMYeB2F326CjYSlKA
rBPuUBQemMc=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFpTCCA42gAwIBAgIUZPYOZXdhaqs7tOqFhLuxibhxkw8wDQYJKoZIhvcNAQEM
BQAwWjELMAkGA1UEBhMCQ04xJTAjBgNVBAoMHFRydXN0QXNpYSBUZWNobm9sb2dp
ZXMsIEluYy4xJDAiBgNVBAMMG1RydXN0QXNpYSBHbG9iYWwgUm9vdCBDQSBHMzAe
Fw0yMTA1MjAwMjEwMTlaFw00NjA1MTkwMjEwMTlaMFoxCzAJBgNVBAYTAkNOMSUw
IwYDVQQKDBxUcnVzdEFzaWEgVGVjaG5vbG9naWVzLCBJbmMuMSQwIgYDVQQDDBtU
cnVzdEFzaWEgR2xvYmFsIFJvb3QgQ0EgRzMwggIiMA0GCSqGSIb3DQEBAQUAA4IC
DwAwggIKAoICAQDAMYJhkuSUGwoqZdC+BqmHO1ES6nBBruL7dOoKjbmzTNyPtxNS
T1QY4SxzlZHFZjtqz6xjbYdT8PfxObegQ2OwxANdV6nnRM7EoYNl9lA+sX4WuDqK
AtCWHwDNBSHvBm3dIZwZQ0WhxeiAysKtQGIXBsaqvPPW5vxQfmZCHzyLpnl5hkA1
nyDvP+uLRx+PjsXUjrYsyUQE49RDdT/VP68czH5GX6zfZBCK70bwkPAPLfSIC7Ep
qq+FqklYqL9joDiR5rPmd2jE+SoZhLsO4fWvieylL1AgdB4SQXMeJNnKziyhWTXA
yB1GJ2Faj/lN03J5Zh6fFZAhLf3ti1ZwA0pJPn9pMRJpxx5cynoTi+jm9WAPzJMs
hH/x/Gr8m0ed262IPfN2dTPXS6TIi/n1Q1hPy8gDVI+lhXgEGvNz8teHHUGf59gX

zhqcD0r83ERoVGjiQTz+LISGNzzNPy+i2+f3VANfWdP3kXjHi3dqFuVJhZBFcnAv
kV34PmVACxmZySYgWmjBNb9Pp1Hx2BErW+Canig7CjoKH8GB5S7wprlppYiU5msT
f9FkPz2ccEblooV7WIQn3MSAPmeamseaMQ4w7OYXQJXZRe0Blqq/DPNL0WP3E1jA
uPP6Z92bfW1K/zJMtSU7/xxnD4UiWQWRkUF3gdCFTIcQcf+eQxuulXUtgQIDAQAB
o2MwYTAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFEDk5PIj7zjKsK5Xf/Ih
MBY027ySMB0GA1UdDgQWBBRA5OTyI+84yrCuV3/yITAWNNu8kjAOBgNVHQ8BAf8E
BAMCAQYwDQYJKoZIhvcNAQEMBQADggIBACY7UeFNOPMyGLS0XuFlXsSUT9SnYaP4
wM8zAQLpw6o1D/GUE3d3NZ4tVlFEbuHGLige/9rsR82XRBf34EzC4Xx8MnpmyFq2
XFNFV1pF1AWZLy4jVe5jaN/TG3inEpQGAHUNcoTpLrxaatXeL1nHo+zSh2bbt1S1
JKv0Q3jbSwTEb93mPmY+KfJLaHEih6D4sTNjduMNhXJEIlU/HHzp/LgV6FL6qj6j
ITk1dImmasI5+njPtqzn59ZW/yOSLlALqbUHM/Q4X6RJpstlcHboCoWASzY9M/eV
VHUl2qzEc4Jl6VL1XP04lQJqaTDFHApXB64ipCz5xUG3uOyfT0gA+QEEVcys+TIx
xHWVBqB/0Y0n3bOppHKH/lmLmnp0Ft0WpWIp6zqW3IunaFnT63eROfjXy9mPX1on
AX1daBli2MjN9LdyR75bl87yraKZk62Uy5P2EgmVtqvXO9A/EcswFi55gORngS1d
7XB4tmBZrOFdRWOPyN9yaFvqHbgB8X7754qz41SgOAngPN5C8sLtLpvzHzW2Ntjj
gKGLzZlkD8Kqq7HK9W+eQ42EVJmzbsASZthwEPEGNTNDqJwuuhQxzhB/HIbjj9LV
+Hfsm6vxL2PZQl/gZ4FkkfGXL/xuJvYz+NO1+MRiqzFRJQJ6+N1rZdVtTTDIZbpo
FGWsJwt0ivKH
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICVTCCAdygAwIBAgIUTyNkuI6XY57GU4HBdk7LKnQV1tcwCgYIKoZIzj0EAwMw
WjELMAkGA1UEBhMCQ04xJTAjBgNVBAoMHFRydXN0QXNpYSBUZWNobm9sb2dpZXMs
IEluYy4xJDAiBgNVBAMMG1RydXN0QXNpYSBHbG9iYWwgUm9vdCBDQSBHNDAeFw0y
MTA1MjAwMjEwMjJaFw00NjA1MTkwMjEwMjJaMFoxCzAJBgNVBAYTAkNOMSUwIwYD
VQQKDBxUcnVzdEFzaWEgVGVjaG5vbG9naWVzLCBJbmMuMSQwIgYDVQQDDBtUcnVz
dEFzaWEgR2xvYmFsIFJvb3QgQ0EgRzQwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAATx
s8045CVD5d4ZCbuBeaIVXxVjAd7Cq92zphtnS4CDr5nLrBfbK5bKfFJV4hrhPVbw
LxYI+hW8m7tH5j/uqOFMjPXTNvk4XatwmkcN4oFBButJ+bAp3TPsUKV/eSm4IJij
YzBhMA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUpbtKl86zK3+kMd6Xg1mD
pm9xy94wHQYDVR0OBBYEFKW7SpfOsyt/pDHel4NZg6ZvccveMA4GA1UdDwEB/wQE
AwIBBjAKBggqhkjOPQQDAwNnADBkAjBe8usGzEkxn0AAbbd+NvBNEU/zy4k6LHiR
UKNbwMp1JvK/kF0LgoxgKJ/GcJpo5PECMFxYDlZ2z1jD1xCMuo6u47xkdUfFVZDj
/bpV6wfEU6s3qe4hsiFbYI89MvHVI5TWWA==
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICMTCCAbegAwIBAgIUNnThTXxlE8msg1UloD5Sfi9QaMcwCgYIKoZIzj0EAwMw
WDELMAkGA1UEBhMCQ04xJTAjBgNVBAoTHFRydXN0QXNpYSBUZWNobm9sb2dpZXMs
IEluYy4xIjAgBgNVBAMTGVRydXN0QXNpYSBUTFMgRUNDIFJvb3QgQ0EwHhcNMjQw
NTE1MDU0MTU2WhcNNDQwNTE1MDU0MTU1WjBYMQswCQYDVQQGEwJDTjElMCMGA1UE
ChMcVHJ1c3RBc2lhIFRlY2hub2xvZ2llcywgSW5jLjEiMCAGA1UEAxMZVHJ1c3RB
c2lhIFRMUyBFQ0MgUm9vdCBDQTB2MBAGByqGSM49AgEGBSuBBAAiA2IABLh/pVs/
AT598IhtrimY4ZtcU5nb9wj/1WrgjstEpvDBjL1P1M7UiFPoXlfXTr4sP/MSpwDp
guMqWzJ8S5sUKZ74LYO1644xST0mYekdcouJtgq7nDM1D9rs3qlKH8kzsaNCMEAw
DwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQULIVTu7FDzTLqnqOH/qKYqKaT6RAw
DgYDVR0PAQH/BAQDAgEGMAoGCCqGSM49BAMDA2gAMGUCMFRH18MtYYZI9HlaVQ01

L18N9mdsd0AaRuf4aFtOJx24mH1/k78ITcTaRTChD15KeAIxAKORh/IRM4PDwYqR
OkwrULG9IpRdNYlzg8WbGf60oenUoWa2AaU2+dhoYSi3dOGiMQ==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFgDCCA2igAwIBAgIUHBjYz+VTPyI1RlNUJDxsR9FcSpwwDQYJKoZIhvcNAQEM
BQAwWDELMAkGA1UEBhMCQ04xJTAjBgNVBAoTHFRydXN0QXNpYSBUZWNobm9sb2dp
ZXMsIEluYy4xIjAgBgNVBAMTGVRydXN0QXNpYSBUTFMgUlNBIFJvb3QgQ0EwHhcN
MjQwNTE1MDU0MTU3WhcNNDQwNTE1MDU0MTU2WjBYMQswCQYDVQQGEwJDTjElMCMG
A1UEChMcVHJ1c3RBc2lhIFRlY2hub2xvZ2llcywgSW5jLjEiMCAGA1UEAxMZVHJ1
c3RBc2lhIFRMUyBSU0EgUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCC
AgoCggIBAMMWuBtqpERz5dZO9LnPWwvB0ZqB9WOwj0PBuwhaGnrhB3YmH49pVr7+
NmDQDIPNlOrnxS1cLwUWAp4KqC/lYCZUlviYQB2srp10Zy9U+5RjmOMmSoPGlbYJ
Q1DNDX3eRA5gEk9bNb2/mThtfWza4mhzH/kxpRkQcwUqwzIZheo0qt1CHjCNP561
HmHVb70AcnKtEj+qpklz8oYVlQwQX1Fkzv93uMltrOXVmPGZLmzjyUT5tUMnCE32
ft5EebuyjBza00tsLtbDeLdM1aTk2tyKjg7/D8OmYCYozza/+lcK7Fs/6TAWe8Tb
xNRkoDD75f0dcZLdKY9BWN4ArTr9PXwaqLEX8E40eFgl1oUh63kd0Nyrz2I8sMeX
i9bQn9P+PN7F4/w6g3CEIR0JwqH8uyghZVNgepBtljhb//HXeltt08lwSUq6HTrQ
UNoyIBnkiz/r1RYmNzz7dZ6wB3C4FGB33PYPXFIKvF1tjVEK2sUYyJtt3LCDs3+j
TnhMmCWr8n4uIF6CFabW2I+s5c0yhsj55NqJ4js+k8UTav/H9xj8Z7XvGCxUq0DT
bE3txci3OE9kxJRMT6DNrqXGJyV1J23G2pyOsAWZ1SgRxSHUuPzHlqtKZFlhaxP8
S8ySpg+kUb8OWJDZgoM5pl+z+m6Ss80zDoWo8SnTq1mt1tve1CuBAgMBAAGjQjBA
MA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFLgHkXlcBvRG/XtZylomkadFK/hT
MA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQwFAAOCAgEAIZtqBSBdGBanEqT3
Rz/NyjuujsCCztxIJXgXbODgcMTWltnZ9r96nBO7U5WS/8+S4PPFJzVXqDuiGev4
iqME3mmL5Dw8veWv0BIb5Ylrc5tvJQJLkIKvQMKtuppgJFqBTQUYo+IzeXoLH5Pt
7DlK9RME7I10nYEKqG/odv6LTytpEoYKNDbdgptvT+Bz3Ul/KD7JO6NXBNiT2Twp
2xIQaOHEibgGIOcberyxk2GaGUARtWqFVwHxtlotJnMnlvm5P1vQiJ3koP26TpUJ
g3933FEFlJ0gcXax7PqJtZwuhfG5WyRasQmr2soaB82G39tp27RIGAAtvKLEiUUj
pQ7hRGU+isFqMB3iYPg6qocJQrmBktwliJiJ8Xw18WLK7nn4GS/+X/jbh87qqA8M
pugLoDzga5SYnH+tBuYc6kIQX+ImFTw3OffXvO645e8D7r0i+yiGNFjEWn9hongP
XvPKnbwbPKfILfanIhHKA9jnZwqKDss1jjQ52MjqjZ9k4DewbNfFj8GQYSbbJIwe
SsCI3zWQzj8C9GRh3sfIB5XeMhg6j6JCQCTl1jNdfK7vsU1P1FeQNWrcrgSXSYk0
ly4wBOeY99sLAZDBHwo/+ML+TvrbmnNzFrwFuHnYWa8G5z9nODmxfKuU4CkUpijy
323imttUQ/hHWKNddBWcwauwxzQ=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF2jCCA8KgAwIBAgIMBfcOhtpJ80Y1LrqyMA0GCSqGSIb3DQEBCwUAMIGIMQsw
CQYDVQQGEwJVUzERMA8GA1UECAwISWxsaW5vaXMxEDAOBgNVBAcMB0NoaWNhZ28x
ITAfBgNVBAoMGFRydXN0d2F2ZSBIb2xkaW5ncywgSW5jLjExMC8GA1UEAwwoVHJ1
c3R3YXZlIEdsb2JhbCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0xNzA4MjMx
OTM0MTJaFw00MjA4MjMxOTM0MTJaMIGIMQswCQYDVQQGEwJVUzERMA8GA1UECAwI
SWxsaW5vaXMxEDAOBgNVBAcMB0NoaWNhZ28xITAfBgNVBAoMGFRydXN0d2F2ZSBI
b2xkaW5ncywgSW5jLjExMC8GA1UEAwwoVHJ1c3R3YXZlIEdsb2JhbCBDZXJ0aWZp
Y2F0aW9uIEF1dGhvcml0eTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIB

ALldUShLPDeS0YLOvR29zd24q88KPuFd5dyqCblXAj7mY2Hf8g+CY66j96xz0Xzn
swuvCAAJWX/NKSqIk4cXGIDtiLK0thAfLdZfVaITXdHG6wZWiYj+rDKd/VzDBcdu
7oaJuogDnXIhhpCujwOl3J+IKMujkkkP7NAP4m1ET4BqstTnoApTAbqOl5F2brz8
1Ws25kCI1nsvXwXoLG0R8+eyvpJETNKXpP7ScoFDB5zpET71ixpZfR9oWN0EACyW
80OzfpgZdNmcc9kYvkHHNHnZ9GLCQ7mzJ7Aiy/k9UscwR7PJPrhq4ufogXBeQotP
JqX+OsIgbrv4Fo7NDKm0G2x2EOFYeUY+VM6AqFcJNykbmROPDMjWLBz7BegIlT1l
RtzuzWniTY+HKE40Cz7PFNm73bZQmq131BnW2hqIyE4bJ3XYsgjxroMwuREOzYfw
hI0Vcnyh78zyiGG69Gm7DIwLdVcEuE4qFC49DxweMqZiNu5m4iK4BUBjECLzMx10
coos9TkpoNPnG4CELcU9402x/RpvumUHO1jsQkUm+9jaJXLE9gCxInm943xZYkqc
BW89zubWR2OZxiRvchLIrH+QtAuRcOi35hYQcRfO3gZPSEF9NUqjifLJS3tBEW1n
twiYTOURGa5CgNz7kAXU+FDKvuStx8KU1xad5hePrzb7AgMBAAGjQjBAMA8GA1Ud
EwEB/wQFMAMBAf8wHQYDVR0OBBYEFJngGWcNYtt2s9o9uFvo/ULSMQ6HMA4GA1Ud
DwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAmHNw4rDT7TnsTGDZqRKGFx6W
0OhUKDtkLSGm+J1WE2pIPU/HPinbbViDVD2HfSMF1OQc3Og4ZYbFdada2zUFvXfe
uyk3QAUHw5RSn8pk3fEbK9xGChACMf1KaA0HZJDmHvUqoai7PF35owgLEQzxPy0Q
lG/+4jSHg9bP5Rs1bdID4bANqKCqRieCNqcVtgimQlRXtpla4gt5kNdXElE1GYhB
aCXUNxeEFfsBctyV3lImIJgm4nb1J2/6ADtKYdkNy1GTKv0WBpanI5ojSP5RvbbE
sLFUzt5sQa0WZ37b/TjNuThOssFgy50X31ieemKyJo90lZvkWx3SD92YHJtZuSPT
MaCm/zjdzyBP6VhWOmfD0faZmZ26NraAL4hHT4a/RDqA5Dccprrql5gR0IRiR2Qe
qu5AvzSxnI9O4fKSTx+O856X3vOmeWqJcU9LJxdI/uz0UA9PSX3MReO9ekDFQdxh
VicGaeVyQYHTtgGJoC86cnn+OjC/QezHYj6RS8fZMXZC+fc8Y+wmjHMMfRod6qh8
h6jCJ3zhM0EPz8/8AKAigJ5Kp28AsEFFtyLKaEjFQqKu3R3y4G5OBVixwJAWKqQ9
EEC+j2Jjg6mcgn0tAumDMHzLJ8n9HmYAsC7TIS+OMxZsmO0QqAfWzJPP29FpHOTK
yeC2nOnOcXHebD8WpHk=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICYDCCAgegAwIBAgIMDWpfCD8oXD5Rld9dMAoGCCqGSM49BAMCMIGRMQswCQYD
VQQGEwJVUzERMA8GA1UECBMISWxsaW5vaXMxEDAOBgNVBAcTB0NoaWNhZ28xITAf
BgNVBAoTGFRydXN0d2F2ZSBIb2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3
YXZlIEdsb2JhbCBFQ0MgUDI1NiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0x
NzA4MjMxOTM1MTBaFw00MjA4MjMxOTM1MTBaMIGRMQswCQYDVQQGEwJVUzERMA8G
A1UECBMISWxsaW5vaXMxEDAOBgNVBAcTB0NoaWNhZ28xITAfBgNVBAoTGFRydXN0
d2F2ZSBIb2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3YXZlIEdsb2JhbCBF
Q0MgUDI1NiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTBZMBMGByqGSM49AgEGCCqG
SM49AwEHA0IABH77bOYj43MyCMpg5lOcunSNGLB4kFKA3TjASh3RqMyTpJcGOMoN
FWLGjgEqZZ2q3zSRLoHB5DOSMcT9CTqmP62jQzBBMA8GA1UdEwEB/wQFMAMBAf8w
DwYDVR0PAQH/BAUDAwcGADAdBgNVHQ4EFgQUo0EGrJBt0UrrdaVKEJmzsaGLSvcw
CgYIKoZIzj0EAwIDRwAwRAIgB+ZU2g6gWrKuEZ+Hxbb/ad4lvvigtwjzRM4q3wgh
DDcCIC0mA6AFvWvR9lz4ZcyGbbOcNEhjhAnFjXca4syc4XR7
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICnTCCAiSgAwIBAgIMCL2Fl2yZJ6SAaEc7MAoGCCqGSM49BAMDMIGRMQswCQYD
VQQGEwJVUzERMA8GA1UECBMISWxsaW5vaXMxEDAOBgNVBAcTB0NoaWNhZ28xITAf
BgNVBAoTGFRydXN0d2F2ZSBIb2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3

YXZlIEdsb2JhbCBFQ0MgUDM4NCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0x
NzA4MjMxOTM2NDNaFw00MjA4MjMxOTM2NDNaMIGRMQswCQYDVQQGEwJVUzERMA8G
A1UECBMISWxsaW5vaXMxEDAOBgNVBAcTB0NoaWNhZ28xITAfBgNVBAoTGFRydXN0
d2F2ZSBIb2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3YXZlIEdsb2JhbCBF
Q0MgUDM4NCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTB2MBAGByqGSM49AgEGBSuB
BAAiA2IABGvaDXU1CDFHBa5FmVXxERMuSvgQMSOjfoPTfygIOiYaOs+Xgh+AtycJ
j9GOMMQKmw6sWASr9zZ9lCOkmwqKi6vr/TklZvFe/oyujUF5nQlgziip04pt89ZF
1PKYhDhloKNDMEEwDwYDVR0TAQH/BAUwAwEB/zAPBgNVHQ8BAf8EBQMDBwYAMB0G
A1UdDgQWBBRVqYSJ0sEyvRjLbKYHTsjnnb6CkDAKBggqhkjOPQQDAwNnADBkAjA3
AZKXRRJ+oPM+rRk6ct30UJMDEr5E0k9BpIycnR+j9sKS50gU/k6bpZFXrsY3crsC
MGclCrEMXu6pY5Jv5ZAL/mYiykf9ijH3g/56vxC+GCsej/YpHpRZ744hN8tRmKVu
Sw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFszCCA5ugAwIBAgIUEwLV4kBMkkaGFmddtLu7sms+/BMwDQYJKoZIhvcNAQEL
BQAwYTELMAkGA1UEBhMCVE4xNzA1BgNVBAoMLkFnZW5jZSBOYXRpb25hbGUgZGUg
Q2VydGlmaWNhdGlvbiBFbGVjdHJvbmlxdWUxGTAXBgNVBAMMEFR1blRydXN0IFJv
b3QgQ0EwHhcNMTkwNDI2MDg1NzU2WhcNNDQwNDI2MDg1NzU2WjBhMQswCQYDVQQG
EwJUTjE3MDUGA1UECgwuQWdlbmNlIE5hdGlvbmFsZSBkZSBDZXJ0aWZpY2F0aW9u
IEVsZWN0cm9uaXF1ZTEZMBcGA1UEAwwQVHVuVHJ1c3QgUm9vdCBDQTCCAiIwDQYJ
KoZIhvcNAQEBBQADggIPADCCAgoCggIBAMPN0/y9BFPdDCA61YguBUtB9YOCfvdZ
n56eY+hz2vYGqU8ftPkLHzmMmiDQfgbU7DTZhrx1W4eI8NLZ1KMKsmwb60ksPqxd
2JQDoOw05TDENX37Jk0bbjBU2PWARZw5rZzJJQRNmpA+TkBuimvNKWfGzC3gdOgF
VwpIUPp6Q9p+7FuaDmJ2/uqdHYVy7BG7NegfJ7/Boce7SBbdVtfMTqDhuazb1YMZ
GoXRlJfXyqNlC/M4+QKu3fZnz8k/9YosRxqZbwUN/dAdgjH8KcwAWJeRTIAAHDOF
li/LQcKLEITDCSSJH7UP2dl3RxiSlGBcx5kDPP73lad9UKGAwqmDrViWVSHbhlnU
r8a83YFuB9tgYv7sEG7aaAH0gxupPqJbI9dkxt/con3YS7qC0lH4Zr8GRuR5KiY2
eY8fTpkdso8MDhz/yV3A/ZAQprE38806JG60hZC/gLkMjNWb1sjxVj8agIl6qeIb
MlEsPvLfe/ZdeikZjuXIvTZxi11Mwh0/rViizz1wTaZQmCXcI/m4WEEIcb9PuISg
jwBUFfyRbVinljvrS5YnzWuioYasDXxU5mZMZl+QviGaAkYt5IPCgLnPSz7ofzwB
7I9ezX/SKEIBlYrilz0QIX32nRzFNKHsLA4KUiwSVXAkPcvCFDVDXSdOvsC9qnyW
5/yeYa1E0wCXAgMBAAGjYzBhMB0GA1UdDgQWBBQGmpsfU33x9aTI04Y+oXNZtPdE
ITAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFAaamx9TffH1pMjThj6hc1m0
90QhMA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAqgVutt0Vyb+z
xiD2BkewhpMl0425yAA/l/VSJ4hxyXT968pk21vvHl26v9Hr7lxpuhbI87mP0zYu
QEkHDVneixCwSQXi/5E/S7fdAo74gShczNxtr18UnH1YeA32gAm56Q6XKRm4t+v4
FstVEuTGfbvE7Pi1HE4+Z7/FXxttbUcoqgRYYdZ2vyJ/0Adqp2RT8JeNnYA/u8EH
22Wv5psymsNUk8QcCMNE+3tjEUPRahphanltkE8pjkcFwRJpadbGNjHh/PqAulxP
xOu3Mqz4dWEX1xAZufHSCe96Qp1bWgvUxpVOKs7/B9dPfhgGiPEZtdmYu65xxBzn
dFlY7wyJz4sfdZMaBBSSSFCp61cpABbjNhzI+L/wM9VBD8TMPN3pM0MBkRArHtG5
Xc0yGYuPjCB31yLEQtyEFpslbei0VXF/sHyz03FJuc9SpAQ/3D2gu68zngowYI7b
nV2UqL1g52KAdoGDDIzMMEZJ4gzSqK/rYXHv5yJiqfdcZGyfFoxnNidF9Ql7v/YQ
CvGwjVRDjAS6oz/v4jXH+XTgbzRB0L9zZVcg+ZtnemZoJE6AZb0QmQZZ8mWvuMZH
u/2QeItBcy6vVR/cO5JyboTT0GFMDcx2V+IthSIVNg3rAZ3r2OvEhJn7wAzMMujj
d9qDRIueVSjAi1jTkD5OGwDxFa2DK5o=
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFWjCCA0KgAwIBAgIQT9Irj/VkyDOeTzRYZiNwYDANBgkqhkiG9w0BAQsFADBH
MQswCQYDVQQGEwJDTjERMA8GA1UECgwIVW5pVHJ1c3QxJTAjBgNVBAMMHFVDQSBF
eHRlbmRlZCBWYWxpZGF0aW9uIFJvb3QwHhcNMTUwMzEzMDAwMDAwWhcNMzgxMjMx
MDAwMDAwWjBHMQswCQYDVQQGEwJDTjERMA8GA1UECgwIVW5pVHJ1c3QxJTAjBgNV
BAMMHFVDQSBFeHRlbmRlZCBWYWxpZGF0aW9uIFJvb3QwggIiMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCpCQcoEwKwmeBkqh5DFnpzsZGgdT6o+uM4AHrsiWog
D4vFsJszA1qGxliG1cGFu0/GnEBNyr7uaZa4rYEwmnySBesFK5pI0Lh2PpbIILvS
sPGP2KxFRv+qZ2C0d35qHzwaUnoEPQc8hQ2E0B92CvdqFN9y4zR8V05WAT558aop
O2z6+I9tTcg1367r3CTueUWnhbYFiN6IXSV8l2RnCdm/WhUFhvMJHuxYMjMR83dk
sHYf5BA1FxvyDrFspCqjc/wJHx4yGVMR59mzLC52LqGj3n5qiAno8geK+LLNEOfi
c0CTuwjRP+H8C5SzJe98ptfRr5//lpr1kXuYC3fUfugH0mK1lTnj8/FtDw5lhIpj
VMWAtuCeS31HJqcBCF3RiJ7XwzJE+oJKCmhUfzhTA8ykADNkUVkLo4KRel7sFsLz
KuZi2irbWWIQJUoqgQtHB0MGcIfS+pMRKXpITeuUx3BNr2fVUbGAIAEBtHoIppB/
TuDvB0GHr2qlXov7z1CymlSvw4m6WC31MJixNnI5fkkE/SmnTHnkBVfblLkWU41G
sx2VYVdWf6/wFlthWG82UBEL2KwrlRYaDh8IzTY0ZRBiZtWAXxQgXy0MoHgKaNYs
1+lvK9JKBZP8nm9rZ/+I8U6laUpSNwXqxhaN0sSZ0YIrO7o1dfdRUVjzyAfd5LQD
fwIDAQABo0IwQDAdBgNVHQ4EFgQU2XQ65DA9DfcS3H5aBZ8eNJr34RQwDwYDVR0T
AQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAYYwDQYJKoZIhvcNAQELBQADggIBADaN
l8xCFWQpN5smLNb7rhVpLGsaGvdftvkHTFnq88nIua7Mui563MD1sC3AO6+fcAUR
ap8lTwEpcOPlDOHqWnzcSbvBHiqB9RZLcpHIojG5qtr8nR/zXUACE/xOHAbKsxSQ
VBcZEhrxH9cMaVr2cXj0lH2RC47skFSOvG+hTKv8dGT9cZr4QQehzZHkPJrgmzI5
c6sq1WnIeJEmMX3ixzDx/BR4dxIOE/TdFpS/S2d7cFOFyrC78zhNLJA5wA3CXWvp
4uXViI3WLL+rG761KIcSF3Ru/H38j9CHJrAb+7lsq+KePRXBOy5nAliRn+/4Qh8s
t2j1da3Ptfb/EX3C8CSlrdP6oDyp+l3cpaDvRKS+1ujl5BOWF3sGPjLtx7dCvHaj
2GU4Kzg1USEODm8uNBNA4StnDG1KQTAYI1oyVZnJF+A83vbsea0rWBmirSwiGpWO
vpaQXUJXxPkUAzUrHC1RVwinOt4/5Mi0A3PCwSaAuwtCH60NryZy2sy+s6ODWA2C
xR9GUeOcGMyNm43sSet1UNWMKFnKdDTajAshqx7qG+XH/RU+wBeq+yNuJkbL+vmx
cmtpzyKEC2IPrNkZAJSidjzULZrtBJ4tBmIQN1IchXIbJ+XMxjHsN+xjWZsLHXbM
fjKaiJUINlK73nZfdklJrX+9ZSCyycErdhh2n1ax
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFRjCCAy6gAwIBAgIQXd+x2lqj7V2+WmUgZQOQ7zANBgkqhkiG9w0BAQsFADA9
MQswCQYDVQQGEwJDTjERMA8GA1UECgwIVW5pVHJ1c3QxGzAZBgNVBAMMElVDQSBH
bG9iYWwgRzIgUm9vdDAeFw0xNjAzMTEwMDAwMDBaFw00MDEyMzEwMDAwMDBaMD0x
CzAJBgNVBAYTAkNOMREwDwYDVQQKDAhVbmlUcnVzdDEbMBkGA1UEAwwSVUNBIEds
b2JhbCBHMiBSb290MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAxeYr
b3zvJgUno4Ek2m/LAfmZmqkywiKHYUGRO8vDaBsGxUypK8FnFyIdK+35KYmToni9
kmugow2ifsqTs6bRjDXVdfkX9s9FxeV67HeToI8jrg4aA3++1NDtLnurRiNb/yzm
VHqUwCoV8MmNsHo7JOHXaOIxPAYzRrZUEaalLyJUKlgNAQLx+hVRZ2zA+te2G3/R
VogvGjqNO7uCEeBHANBSh6v7hn4PJGtAnTRnvI3HLYZveT6OqTwXS3+wmeOwcWDc
C/Vkw85DvG1xudLeJ1uK6NjGruFZfc8oLTW4lVYa8bJYS7cSN8h8s+1LgOGN+jIj
tm+3SJUIsUROhYw6AlQgL9+/V087OpAh18EmNVQg7Mc/R+zvWr9LesGtOxdQXGLY
D0tK3Cv6brxzks3sx1DoQZbXqX5t2Okdj4q1uViSukqSKwxW/YDrCPBeKW4bHAyv

j5OJrdu9o54hyokZ7N+1wxrrFv54NkzWbtA+FxyQF2smuvt6L78RHBgOLXMDj6Dl
NaBa4kx1HXHhOThTeEDMg5PXCp6dW4+K5OXgSORIskfNTip1KnvyIvbJvgmRlld6
iIis7nCs+dwp4wwcOxJORNanTrAmyPPZGpeRaOrvjUYG0lZFWJo8DA+DuAUlwznP
O6Q0ibd5Ei9Hxeepl2n8pndntd978XplFeRhVmUCAwEAAaNCMEAwDgYDVR0PAQH/
BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFIHEjMz15DD/pQwIX4wV
ZyF0Ad/fMA0GCSqGSIb3DQEBCwUAA4ICAQATZSL1jiutROTL/7lo5sOASD0Ee/oj
L3rtNtqyzm325p7lX1iPyzcyochltq44PTUbPrw7tgTQvPlJ9Zv3hcU2tsu8+Mg5
1eRfB70VVJd0ysrtT7q6ZHafgbiERUlMjW+i67HM0cOU2kTC5uLqGOiiHycFutfl
1qnN3e92mI0ADs0b+gO3joBYDic/UvuUospeZcnWhNq5NXHzJsBPd+aBJ9J3O5oU
b3n09tDh05S60FdRvScFDcH9yBIw7m+NESsIndTUv4BFFJqIRNow6rSn4+7vW4LV
PtateJLbXDzz2K36uGt/xDYotgIVilQsnLAXc47QN6MUPJiVAAwpBVueSUmxX8fj
y88nZY41F7dXyDDZQVu5FLbowg+UMaeUmMxq67XhJ/UQqAHojhJi6IjMtX9Gl8Cb
EGY4GjZGXyJoPd/JxhMnq1MGrKI8hgZlb7F+sSlEmqO6SWkoaY/X5V+tBIZkbxqg
DMUIYs6Ao9Dz7GjevjPHF1t/gMRMTLGmhIrDO7gJzRSBuhjjVFc2/tsvfEehOjPI
+Vg7RE+xygKJBJYoaMVLuCaJu9YzL1DV/pqJuhgyklTGW+Cd+V7lDSKb9triyCGy
YiGqhkCyLmTTX8jjfhFnRR8F/uOi77Oos/N9j/gMHyIfLXC0uAE0djAA5SN4p1bX
UB+K+wb1whnw0A==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICjzCCAhWgAwIBAgIQXIuZxVqUxdJxVt7NiYDMJjAKBggqhkjOPQQDAzCBiDEL
MAkGA1UEBhMCVVMxEzARBgNVBAgTCk5ldyBKZXJzZXkxFDASBgNVBAcTC0plcnNl
eSBDaXR5MR4wHAYDVQQKExVUaGUgVVNFUlRSVVNUIE5ldHdvcmsxLjAsBgNVBAMT
JVVTRVJUcnVzdCBFQ0MgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMTAwMjAx
MDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBiDELMAkGA1UEBhMCVVMxEzARBgNVBAgT
Ck5ldyBKZXJzZXkxFDASBgNVBAcTC0plcnNleSBDaXR5MR4wHAYDVQQKExVUaGUg
VVNFUlRSVVNUIE5ldHdvcmsxLjAsBgNVBAMTJVVTRVJUcnVzdCBFQ0MgQ2VydGlm
aWNhdGlvbiBBdXRob3JpdHkwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAAQarFRaqflo
I+d61SRvU8Za2EurxtW20eZzca7dnNYMYf3boIkDuAUU7FfO7l0/4iGzzvfUinng
o4N+LZfQYcTxmdwlkWOrfzCjtHDix6EznPO/LlxTsV+zfTJ/ijTjeXmjQjBAMB0G
A1UdDgQWBBQ64QmG1M8ZwpZ2dEl23OA1xmNjmjAOBgNVHQ8BAf8EBAMCAQYwDwYD
VR0TAQH/BAUwAwEB/zAKBggqhkjOPQQDAwNoADBlAjA2Z6EWCNzklwBBHU6+4WMB
zzuqQhFkoJ2UOQIReVx7Hfpkue4WQrO/isIJxOzksU0CMQDpKmFHjFJKS04YcPbW
RNZu9YO6bVi9JNlWSOrvxKJGgYhqOkbRqZtNyWHa0V1Xahg=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF3jCCA8agAwIBAgIQAf1tMPyjylGoG7xkDjUDLTANBgkqhkiG9w0BAQwFADCB
iDELMAkGA1UEBhMCVVMxEzARBgNVBAgTCk5ldyBKZXJzZXkxFDASBgNVBAcTC0pl
cnNleSBDaXR5MR4wHAYDVQQKExVUaGUgVVNFUlRSVVNUIE5ldHdvcmsxLjAsBgNV
BAMTJVVTRVJUcnVzdCBSU0EgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMTAw
MjAxMDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBiDELMAkGA1UEBhMCVVMxEzARBgNV
BAgTCk5ldyBKZXJzZXkxFDASBgNVBAcTC0plcnNleSBDaXR5MR4wHAYDVQQKExVU
aGUgVVNFUlRSVVNUIE5ldHdvcmsxLjAsBgNVBAMTJVVTRVJUcnVzdCBSU0EgQ2Vy
dGlmaWNhdGlvbiBBdXRob3JpdHkwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIK
AoICAQCAEmUXNg7D2wiz0KxXDXbtzSfTTK1Qg2HiqiBNCS1kCdzOiZ/MPans9s/B

3PHTsdZ7NygRK0faOca8Ohm0X6a9fZ2jY0K2dvKpOyuR+OJv0OwWIJAJPuLodMkY
tJHUYmTbf6MG8YgYapAiPLz+E/CHFHv25B+O1ORRxhFnRghRy4YUVD+8M/5+bJz/
Fp0YvVGONaanZshyZ9shZrHUm3gDwFA66Mzw3LyeTP6vBZY1H1dat//O+T23LLb2
VN3I5xI6Ta5MirdcmrS3ID3KfyI0rn47aGYBROcBTkZTmzNg95S+UzeQc0PzMsNT
79uq/nROacdrjGCT3sTHDN/hMq7MkztReJVni+49Vv4M0GkPGw/zJSZrM233bkf6
c0Plfg6lZrEpfDKEY1WJxA3Bk1QwGROs0303p+tdOmw1XNtB1xLaqUkL39iAigmT
Yo61Zs8liM2EuLE/pDkP2QKe6xJMlXzzawWpXhaDzLhn4ugTncxbgtNMs+1b/97l
c6wjOy0AvzVVdAlJ2ElYGn+SNuZRkg7zJn0cTRe8yexDJtC/QV9AqURE9JnnV4ee
UB9XVKg+/XRjL7FQZQnmWEIuQxpMtPAlR1n6BB6T1CZGSlCBst6+eLf8ZxXhyVeE
Hg9j1uliutZfVS7qXMYoCAQlObgOK6nyTJccBz8NUvXt7y+CDwIDAQABo0IwQDAd
BgNVHQ4EFgQUU3m/WqorSs9UgOHYm8Cd8rIDZsswDgYDVR0PAQH/BAQDAgEGMA8G
A1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEMBQADggIBAFzUfA3P9wF9QZllDHPF
Up/L+M+ZBn8b2kMVn54CVVeWFPFSPCeHlCjtHzoBN6J2/FNQwISbxmtOuowhT6KO
VWKR82kV2LyI48SqC/3vqOlLVSoGIG1VeCkZ7l8wXEskEVX/JJpuXior7gtNn3/3
ATiUFJVDBwn7YKnuHKsSjKCaXqeYalltiz8I+8jRRa8YFWSQEg9zKC7F4iRO/Fjs
8PRF/iKz6y+O0tlFYQXBl2+odnKPi4w2r78NBc5xjeambx9spnFixdjQg3IM8WcR
iQycE0xyNN+81XHfqnHd4blsjDwSXWXavVcStkNr/+XeTWYRUc+ZruwXtuhxkYze
Sf7dNXGiFSeUHM9h4ya7b6NnJSFd5t0dCy5oGzuCr+yDZ4XUmFF0sbmZgIn/f3gZ
XHlKYC6SQK5MNyosycdiyA5d9zZbyuAlJQG03RoHnHcAP9Dc1ew91Pq7P8yF1m9/
qS3fuQL39ZeatTXaw2ewh0qpKJ4jjv9cJ2vhsE/zB+4ALtRZh8tSQZXq9EfX7mRB
VXyNWQKV3WKdwrnuWih0hKWbt5DHDAff9Yk2dDLWKMGwsAvgnEzDHNb842m1R0aB
L6KCq9NjRHDEjf8tM7qtj3u1cIiuPhnPQCjY/MiQu12ZIvVS5ljFH4gxQ+6IHdfG
jjxDah2nGN59PRbxYvnKkKj9
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDODCCAiCgAwIBAgIGIAYFFnACMA0GCSqGSIb3DQEBBQUAMDsxCzAJBgNVBAYT
AlJPMREwDwYDVQQKEwhjZXJ0U0lHTjEZMBcGA1UECxMQY2VydFNJR04gUk9PVCBD
QTAeFw0wNjA3MDQxNzIwMDRaFw0zMTA3MDQxNzIwMDRaMDsxCzAJBgNVBAYTAlJP
MREwDwYDVQQKEwhjZXJ0U0lHTjEZMBcGA1UECxMQY2VydFNJR04gUk9PVCBDQTCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALczuX7IJUqOtdu0KBuqV5Do
0SLTZLrTk+jUrIZhQGpgV2hUhE28alQCBf/fm5oqrl0Hj0rDKH/v+yv6efHHrfAQ
UySQi2bJqIirr1qjAOm+ukbuW3N7LBeCgV5iLKECZbO9xSsAfsT8AzNXDe3i+s5d
RdY4zTW2ssHQnIFKquSyAVwdj1+ZxLGt24gh65AIgoDzMKND5pCCrlUoSe1b16kQ
OA7+j0xbm0bqQfWwCHTD0IgztnzXdN/chNFDDnU5oSVAKOp4yw4sLjmdjItuFhwv
JoIQ4uNllAoEwF73XVv4EOLQunpL+943AAAaWyjj0pxzPjKHmKHJUS/X3qwzs08C
AwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAcYwHQYDVR0O
BBYEFOCMm9slSbPxfIbWskKHC9BroNnkMA0GCSqGSIb3DQEBBQUAA4IBAQA+0hyJ
LjX8+HXd5n9liPRyTMks1zJO890ZeUe9jjtbkw9QSSQTaxQGcu8J06Gh40CEyecY
MnQ8SG4Pn0vU9x7Tk4ZkVJdjclDVVc/6IJMCopvDI5NOFlV2oHB5bc0hH88vLbwZ
44gx+FkagQnIl6Z0x2DEW8xXjrJ1/RsCCdtZb3KTafcxQdaIOL+Hsr0Wefmq5L6I
Jd1hJyMctTEHBDa0GpC9oHRxUIltvBTjD4au8as+x6AJzKNI0eDbZOeStc+vckNw
i/nDhDwTqn6Sm1dTk/pwwpEOMfmbZ13pljheX7NzTogVZ96edhBiIL5VaZVDADlN
9u6wWk5JRFRYX0KD
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFRzCCAy+gAwIBAgIJEQA0tk7GNi02MA0GCSqGSIb3DQEBCwUAMEExCzAJBgNV
BAYTAlJPMRQwEgYDVQQKEwtDRVJUU0lHTiBTQTEcMBoGA1UECxMTY2VydFNJR04g
Uk9PVCBDQSBHMjAeFw0xNzAyMDYwOTI3MzVaFw00MjAyMDYwOTI3MzVaMEExCzAJ
BgNVBAYTAlJPMRQwEgYDVQQKEwtDRVJUU0lHTiBTQTEcMBoGA1UECxMTY2VydFNJ
R04gUk9PVCBDQSBHMjCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAMDF
dRmRfUR0dIf+DjuW3NgBFszuY5HnC2/OOwppGnzC46+CjobXXo9X69MhWf05N0Iw
vlDqtg+piNguLWkh59E3GE59kdUWX2tbAMI5Qw02hVK5U2UPHULlj88F0+7cDBrZ
uIt4ImfkabBoxTzkbFpG583H+u/E7Eu9aqSs/cwoUe+StCmrqzWaTOTECMYmzPhp
n+Sc8CnTXPnGFiWeI8MgwT0PPzhAsP6CRDiqWhqKa2NYOLQV07YRaXseVO6MGiKs
cpc/I1mbySKEwQdPzH/iV8oScLumZfNpdWO9lfsbl83kqK/20U6o2YpxJM02PbyW
xPFsqa7lzw1uKA2wDrXKUXt4FMMgL3/7FFXhEZn91QqhngLjYl/rNUssuHLoPj1P
rCy7Lobio3aP5ZMqz6WryFyNSwb/EkaseMsUBzXgqd+L6a8VTxaJW732jcZZroiF
DsGJ6x9nxUWO/203Nit4ZoORUSs9/1F3dmKh7Gc+PoGD4FapUB8fepmrY7+EF3fx
DTvf95xhszWYijqy7DwaNz9+j5LP2RIUZNoQAhVB/0/E6xyjyfqZ90bp4RjZsbgy
LcsUDFDYg2WD7rlcz8sFWkz6GZdr1l0T08JcVLwyc6B49fFtHsufpaafItzRUZ6C
eWRgKRM+o/1Pcmqr4tTluCRVLERLiohEnMqE0yo7AgMBAAGjQjBAMA8GA1UdEwEB
/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBSCIS1mxteg4BXrzkwJ
d8RgnlRuAzANBgkqhkiG9w0BAQsFAAOCAgEAYN4auOfyYILVAzOBywaK8SJJ6ejq
kX/GM15oGQOGO0MBzwdw5AgeZYWR5hEit/UCI46uuR59H35s5r0l1ZUa8gWmr4UC
b6741jH/JclKyMeKqdmfS0mbEVeZkkMR3rYzpMzXjWR91M08KCy0mpbqTfXERMQl
qiCA2ClV9+BB/AYm/7k29UMUA2Z44RGx2iBfRgB4ACGlHgAoYXhvqAEBj500mv/0
OJD7uNGzcgbJceaBxXntC6Z58hMLnPddDnskk7RI24Zf3lCGeOdA5jGokHZwYa+c
NywRtYK3qq4kNFtyDGkNzVmf9nGvnAvRCjj5BiKDUyUM/FHE5r7iOZULJK2v0ZXk
ltd0ZGtxTgI8qoXzIKNDOXZbbFD+mpwUHmUUihW9o4JFWklWatKcsWMy5WHgUyIO
pwpJ6st+H6jiYoD2EEVSmAYY3qXNL3+q1Ok+CHLsIwMCPKaq2LxndD0UF/tUSxfj
03k9bWtJySgOLnRQvwzZRjoQhsmnP+mg7H/rpXdYaXHmgwo38oZJar55CJD2AhZk
PuXaTH4MNMn5X7azKFGnpyuqSfqNZSlO42sTp5SjLVFteAxEy9/eCG/Oo2Sr05WE
1LlSVHJ7liXMvGnjSG4N0MedJ5qq+BOS3R7fY581qRY27Iy4g/Q9iY/NtBde17MX
QRBdJ3NghVdJIgc=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICQDCCAeWgAwIBAgIMAVRI7yH9l1kN9QQKMAoGCCqGSM49BAMCMHExCzAJBgNV
BAYTAkhVMREwDwYDVQQHDAhCdWRhcGVzdDEWMBQGA1UECgwNTWljcm9zZWMgTHRk
LjEXMBUGA1UEYQwOVkFUSFUtMjM1ODQ0OTcxHjAcBgNVBAMMFWUtU3ppZ25vIFJv
b3QgQ0EgMjAxNzAeFw0xNzA4MjIxMjA3MDZaFw00MjA4MjIxMjA3MDZaMHExCzAJ
BgNVBAYTAkhVMREwDwYDVQQHDAhCdWRhcGVzdDEWMBQGA1UECgwNTWljcm9zZWMg
THRkLjEXMBUGA1UEYQwOVkFUSFUtMjM1ODQ0OTcxHjAcBgNVBAMMFWUtU3ppZ25v
IFJvb3QgQ0EgMjAxNzBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABJbcPYrYsHtv
xie+RJCxs1YVe45DJH0ahFnuY2iyxl6H0BVIHqiQrb1TotreOpCmYF9oMrWGQd+H
Wyx7xf58etqjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0G
A1UdDgQWBBSHERUI0arBeAyxr87GyZDvvzAEwDAfBgNVHSMEGDAWgBSHERUI0arB
eAyxr87GyZDvvzAEwDAKBggqhkjOPQQDAgNJADBGAiEAtVfd14pVCzbhhkT61Nlo
jbjcI4qKDdQvfepz7L9NbKgCIQDLpbQS+ue16M9+k/zzNY9vTlp8tLxOsvxyqltZ
+efcMQ==
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFsDCCA5igAwIBAgIQFci9ZUdcr7iXAF7kBtK8nTANBgkqhkiG9w0BAQUFADBe
MQswCQYDVQQGEwJUVzEjMCEGA1UECgwaQ2h1bmdod2EgVGVsZWNvbSBDby4sIEx0
ZC4xKjAoBgNVBAsMIWVQS0kgUm9vdCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAe
Fw0wNDEyMjAwMjMxMjdaFw0zNDEyMjAwMjMxMjdaMF4xCzAJBgNVBAYTAlRXMSMw
IQYDVQQKDBpDaHVuZ2h3YSBUZWxlY29tIENvLiwgTHRkLjEqMCgGA1UECwwhZVBL
SSBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIICIjANBgkqhkiG9w0BAQEF
AAOCAg8AMIICCgKCAgEA4SUP7o3biDN1Z82tH306Tm2d0y8U82N0ywEhajfqhFAH
SyZbCUNsIZ5qyNUD9WBpj8zwIuQf5/dqIjG3LBXy4P4AakP/h2XGtRrBp0xtInAh
ijHyl3SJCRImHJ7K2RKilTza6We/CKBk49ZCt0Xvl/T29de1ShUCWH2YWEtgvM3X
DZoTM1PRYfl61dd4s5oz9wCGzh1NlDivqOx4UXCKXBCDUSH3ET00hl7lSM2XgYI1
TBnsZfZrxQWh7kcT1rMhJ5QQCtkkO7q+RBNGMD+XPNjX12ruOzjjK9SXDrkb5wdJ
fzcq+Xd4z1TtW0ado4AOkUPB1ltfFLqfpo0kR0BZv3I4sjZsN/+Z0V0OWQqraffA
sgRFelQArr5T9rXn4fg8ozHSqf4hUmTFpmfwdQcGlBSBVcYn5AGPF8Fqcde+S/uU
WH1+ETOxQvdibBjWzwloPn9s9h6PYq2lY9sJpx8iQkEeb5mKPtf5P0B6ebClAZLS
nT0IFaUQAS2zMnaolQ2zepr7BxB4EW/hj8e6DyUadCrlHJhBmd8hh+iVBmoKs2pH
dmX2Os+PYhcZewoozRrSgx4hxyy/vv9haLdnG7t4TY3OZ+XkwY63I2binZB1NJip
NiuKmpS5nezMirH4JYlcWrYvjB9teSSnUmjDhDXiZo1jDiVN1Rmy5nk3pyKdVDEC
AwEAAaNqMGgwHQYDVR0OBBYEFB4M97Zn8uGSJglFwFU5Lnc/QkqiMAwGA1UdEwQF
MAMBAf8wOQYEZyoHAAQxMC8wLQIBADAJBgUrDgMCGgUAMAcGBWcqAwAABBRFsMLH
ClZ87lt4DJX5GFPBphzYEDANBgkqhkiG9w0BAQUFAAOCAgEACbODU1kBPpVJufGB
uvl2ICO1J2B01GqZNF5sAFPZn/KmsSQHRGoqxqWOeBLoR9lYGxMqXnmbnwoqZ6Yl
PwZpVnPDimZI+ymBV3QGypzqKOg4ZyYr8dW1P2WT+DZdjo2NQCCHGervJ8A9tDkP
JXtoUHRVnAxZfVo9QZQlUgjgRywVMRnVvwdVxrsStZf0X4OFunHB2WyBEXYKCrC/
gpf36j36+uwtqSiUO1bd0lEursC9CBWMd1I0ltabrNMdjmEPNXubrjlpC2JgQCA2
j6/7Nu4tCEoduL+bXPjqpRugc6bY+G7gMwRfaKonh+3ZwZCc7b3jajWvY9+rGNm6
5ulK6lCKD2GTHuItGeIwlDWSXQ62B68ZgI9HkFFLLk3dheLSClIKF5r8GrBQAuUB
o2M3IUxExJtRmREOc5wGj1QupyheRDmHVi03vYVElOEMSyycw5KFNGHLD7ibSkNS
/jQ6fbjpKdx2qcgw+BRxgMYeNkh0IkFch4LoGHGLQYlE535YW6i4jRPpp2zDR+2z
Gp1iro2C6pSe3VkQw63d4k3jMdXH7OjysP6SHhYKGvzZ8/gntsm+HbRsZJB/9OTE
W9c3rkIO3aQab3yIVMUWbuF6aC74Or8NpDyJO3inTmODBCEIZ43ygknQW/2xzQ+D
hNQ+IIX3Sj0rnP0qCglN6oH4EZw=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICKzCCAbGgAwIBAgIKe3G2gla4EnycqDAKBggqhkjOPQQDAzBaMQswCQYDVQQG
EwJVUzETMBEGA1UECxMKZW1TaWduIFBLSTEUMBIGA1UEChMLZU11ZGhyYSBJbmMx
IDAeBgNVBAMTF2VtU2lnbiBFQ0MgUm9vdCBDQSAtIEMzMB4XDTE4MDIxODE4MzAw
MFoXDTQzMDIxODE4MzAwMFowWjELMAkGA1UEBhMCVVMxEzARBgNVBAsTCmVtU2ln
biBQS0kxFDASBgNVBAoTC2VNdWRocmEgSW5jMSAwHgYDVQQDExdlbVNpZ24gRUND
IFJvb3QgQ0EgLSBDMzB2MBAGByqGSM49AgEGBSuBBAAiA2IABP2lYa57JhAd6bci
MK4G9IGzsUJxlTm801Ljr6/58pc1kjZGDoeVjbk5Wum739D+yAdBPLtVb4Ojavti
sIGJAnB9SMVK4+kiVCJNk7tCDK93nCOmfddhEc5lx/h//vXyqaNCMEAwHQYDVR0O
BBYEFPtaSNCAIEDyqOkAB2kZd6fmw/TPMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMB
Af8EBTADAQH/MAoGCCqGSM49BAMDA2gAMGUCMQC02C8Cif22TGK6Q04ThHK1rt0c

3ta13FaPWEBaLd4gTCKDypOofu4SQMfWh0/434UCMBwUZOR8loMRnLDRWmFLpg9J
0wD8ofzkpf9/rdcw0Md3f76BB1UwUCAU9Vc4CqgxUQ==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICTjCCAdOgAwIBAgIKPPYHqWhwDtqLhDAKBggqhkjOPQQDAzBrMQswCQYDVQQG
EwJJTjETMBEGA1UECxMKZW1TaWduIFBLSTElMCMGA1UEChMcZU11ZGhyYSBUZWNo
bm9sb2dpZXMgTGltaXRlZDEgMB4GA1UEAxMXZW1TaWduIEVDQyBSb290IENBIC0g
RzMwHhcNMTgwMjE4MTgzMDAwWhcNNDMwMjE4MTgzMDAwWjBrMQswCQYDVQQGEwJJ
TjETMBEGA1UECxMKZW1TaWduIFBLSTElMCMGA1UEChMcZU11ZGhyYSBUZWNobm9s
b2dpZXMgTGltaXRlZDEgMB4GA1UEAxMXZW1TaWduIEVDQyBSb290IENBIC0gRzMw
djAQBgcqhkjOPQIBBgUrgQQAIgNiAAQjpQy4LRL1KPOxst3iAhKAnjlfSU2fySU0
WXTsuwYc58Byr+iuL+FBVIcUqEqy6HyC5ltqtdyzdc6LBtCGI79G1Y4PPwT01xyS
fvalY8L1X44uT6EYGQIrMgqCZH0Wk9GjQjBAMB0GA1UdDgQWBBR8XQKEE9TMipuB
zhccLikenEhjQjAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAKBggq
hkjOPQQDAwNpADBmAjEAvvNhzwIQHWSVB7gYboiFBS+DCBeQyh+KTOgNG3qxrdWB
CUfvO6wIBHxcmbHtRwfSAjEAnbpV/KlK6O3t5nYBQnvI+GDZjVGLVTv7jHvrZQnD
+JbNR6iC8hZVdyR+EhCVBCyj
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDczCCAlugAwIBAgILAK7PALrEzzL4Q7IwDQYJKoZIhvcNAQELBQAwVjELMAkG
A1UEBhMCVVMxEzARBgNVBAsTCmVtU2lnbiBQS0kxFDASBgNVBAoTC2VNdWRocmEg
SW5jMRwwGgYDVQQDExNlbVNpZ24gUm9vdCBDQSAtIEMxMB4XDTE4MDIxODE4MzAw
MFoXDTQzMDIxODE4MzAwMFowVjELMAkGA1UEBhMCVVMxEzARBgNVBAsTCmVtU2ln
biBQS0kxFDASBgNVBAoTC2VNdWRocmEgSW5jMRwwGgYDVQQDExNlbVNpZ24gUm9v
dCBDQSAtIEMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAz+upufGZ
BczYKCFK83M0UYRWEPWgTywS4/oTmifQz/l5GnRfHXk5/Fv4cI7gklL35CX5VIPZ
HdPIWoU/Xse2B+4+wM6ar6xWQio5JXDWv7V7Nq2s9nPczdcdioOl+yuQFTdrHCZH
3DspVpNqs8FqOp099cGXOFgFixwR4+S0uF2FHYP+eF8LRWgYSKVGczQ7/g/IdrvH
GPMF0Ybzhe3nudkyrVWIzqa2kbBPrH4VI5b2P/AgNBbeCsbEBEV5f6f9vtKppa+c
xSMq9zwhbL2vj07FOrLzNBL834AaSaTUqZX3noleoomslMuoaJuvimUnzYnu3Yy1
aylwQ6BpC+S5DwIDAQABo0IwQDAdBgNVHQ4EFgQU/qHgcB4qAzlSWkK+XJGFehiq
TbUwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEL
BQADggEBAMJKVvoVIXsoounlHfv4LcQ5lkFMOycsxGwYFYDGrK9HWS8mC+M2sO87
/kOXSTKZEhVb3xEp/6tT+LvBeA+snFOvV71ojD1pM/CjoCNjO2RnIkSt1XHLVip4
kqNPEjE2NuLe/gDEo2APJ62gsIq1NnpSob0n9CAnYuhNlCQT5AoE6TyrLshDCUrG
YQTlSTR+08TI9Q/Aqum6VF7zYytPT1DU/rl7mYw9wC68AivTxEDkigcxHpvOJpkT
+xHqmiIMERnHXhuBUDDIlhJu58tBf5E7oke3VIAb3ADMmpDqw8NQBmIMMMAVSKeo
WXzhriKi4gp6D/piq1JM4fHfyr6DDUI=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDlDCCAnygAwIBAgIKMfXkYgxsWO3W2DANBgkqhkiG9w0BAQsFADBnMQswCQYD
VQQGEwJJTjETMBEGA1UECxMKZW1TaWduIFBLSTElMCMGA1UEChMcZU11ZGhyYSBU

ZWNobm9sb2dpZXMgTGltaXRlZDEcMBoGA1UEAxMTZW1TaWduIFJvb3QgQ0EgLSBH
MTAeFw0xODAyMTgxODMwMDBaFw00MzAyMTgxODMwMDBaMGcxCzAJBgNVBAYTAklO
MRMwEQYDVQQLEwplbVNpZ24gUEtJMSUwIwYDVQQKExxlTXVkaHJhIFRlY2hub2xv
Z2llcyBMaW1pdGVkMRwwGgYDVQQDExNlbVNpZ24gUm9vdCBDQSAtIEcxMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAk0u76WaK7p1b1TST0Bsew+eeuGQz
f2N4aLTNLnF115sgxk0pvLZoYIr3IZpWNVrzdr3YzZr/k1ZLpVkGoZM0Kd0WNHVO
8oG0x5ZOrRkVUkr+PHB1cM2vK6sVmjM8qrOLqs1D/fXqcP/tzxE7lM5OMhbTI0Aq
d7OvPAEsbO2ZLIvZTmmYsvePQbAyeGHWDV/D+qJAkh1cF+ZwPjXnorfCYuKrpDhM
tTk1b+oDafo6VGiFbdbyL0NVHpENDtjVaqSW0RM8LHhQ6DqS0hdW5TUaQBw+jSzt
Od9C4INBdN+jzcKGYEho42kLVACL5HZpIQ15TjQIXhTCzLG3rdd8cIrHhQIDAQAB
o0IwQDAdBgNVHQ4EFgQU++8Nhp6w492pufEhF38+/PB3KxowDgYDVR0PAQH/BAQD
AgEGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAFn/8oz1h31x
PaOfG1vR2vjTnGs2vZupYeveFix0PZ7mddrXuqe8QhfnPZHr5X3dPpzxz5KsbEjM
wiI/aTvFthUvozXGaCocV685743QNcMYDHsAVhzNixl03r4PEuDQqqE/AjSxcM6d
GNYIAwlG7mDgfrbESQRRfXBgvKqy/3lyeqYdPV8q+Mri/Tm3R7nrft8EI6/6nAYH
6ftjk4BAtcZsCjEozgyfz7MjNYBBjWzEN3uBL4ChQEKF6dk4jeihU80Bv2noWgby
RQuQ+q7hv53yrlc8pa6yVvSLZUDp/TGBLPQ5Cdjua6e0ph0VpZj3AYHYhX3zUVxx
iN66zB+Afko=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICDzCCAZWgAwIBAgIUbmq8WapTvpg5Z6LSa6Q75m0c1towCgYIKoZIzj0EAwMw
RzELMAkGA1UEBhMCQ04xHDAaBgNVBAoTE2lUcnVzQ2hpbmEgQ28uLEx0ZC4xGjAY
BgNVBAMTEXZUcnVzIEVDQyBSb290IENBMB4XDTE4MDczMTA3MjY0NFoXDTQzMDcz
MTA3MjY0NFowRzELMAkGA1UEBhMCQ04xHDAaBgNVBAoTE2lUcnVzQ2hpbmEgQ28u
LEx0ZC4xGjAYBgNVBAMTEXZUcnVzIEVDQyBSb290IENBMHYwEAYHKoZIzj0CAQYF
K4EEACIDYgAEZVBKrox5lkqqHAjDo6LN/llWQXf9JpRCux3NCNtzslt188+cToL0
v/hhJoVs1oVbcnDS/dtitN9Ti72xRFhiQgnH+n9bEOf+QP3A2MMrMudwpremIFUd
e4BdS49nTPEQo0IwQDAdBgNVHQ4EFgQUmDnNvtiyjPeyq+GtJK97fKHbH88wDwYD
VR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwCgYIKoZIzj0EAwMDaAAwZQIw
V53dVvHH4+m4SVBrm2nDb+zDfSXkV5UTQJtS0zvzQBm8JsctBp61ezaf9SXUY2sA
AjEA6dPGnlaaKsyh2j/IZivTWJwghfqrkYpwcBE4YGQLYgmRWAD5Tfs0aNoJrSEG
GJTO
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFVjCCAz6gAwIBAgIUQ+NxE9izWRRdt86M/TX9b7wFjUUwDQYJKoZIhvcNAQEL
BQAwQzELMAkGA1UEBhMCQ04xHDAaBgNVBAoTE2lUcnVzQ2hpbmEgQ28uLEx0ZC4x
FjAUBgNVBAMTDXZUcnVzIFJvb3QgQ0EwHhcNMTgwNzMxMDcyNDA1WhcNNDMwNzMx
MDcyNDA1WjBDMQswCQYDVQQGEwJDTjEcMBoGA1UEChMTaVRydXNDaGluYSBDby4s
THRkLjEWMBQGA1UEAxMNdlRydXMgUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEBBQAD
ggIPADCCAgoCggIBAL1VfGHTuB0EYgWgrmy3cLRB6ksDXhA/kFocizuwZotsSKYc
IrrVQJLuM7IjWcmOvFjai57QGfIvWcaMY1q6n6MLsLOaXLoRuBLpDLvPbmyAhykU
AyyNJJrIZIO1aqwTLDPxn9wsYTwaP3BVm60AUn/PBLn+NvqcwBauYv6WTEN+VRS+
GrPSbcKvdmaVayqwlHeFXgQPYh1jdfdr58tbmnDsPmcF8P4HCIDPKNsFxhQnL4Z9
8Cfe/+Z+M0jnCx5Y0ScrUw5XSmXX+6KAYPxMvDVTAWqXcoKv8R1w6Jz1717CbMdH

flqUhSZNO7rrTOiwCcJlwp2dCZtOtZcFrPUGoPc2BX70kLJrxLT5ZOrpGgrIDajt
J8nU57O5q4IikCc9Kuh8kO+8T/3iCiSn3mUkpF3qwHYw03dQ+A0Em5Q2AXPKBlim
0zvc+gRGE1WKyURHuFE5Gi7oNOJ5y1lKCn+8pu8fA2dqWSslYpPZUxlmPCdiKYZN
pGvu/9ROutW04o5IWgAZCfEF2c6Rsffr6TlP9m8EQ5pV9T4FFL2/s1m02I4zhKOQ
UqqzApVg+QxMaPnu1RcN+HFXtSXkKe5lXa/R7jwXC1pDxaWG6iSe4gUH3DRCEpHW
OXSuTEGC2/KmSNGzm/MzqvOmwMVO9fSddmPmAsYiS8GVP1BkLFTltvA8Kc9XAgMB
AAGjQjBAMB0GA1UdDgQWBBRUYnBj8XWEQ1iO0RYgscasGrz2iTAPBgNVHRMBAf8E
BTADAQH/MA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAKbqSSaet
8PFww+SX8J+pJdVrnjT+5hpk9jprUrIQeBqfTNqK2uwcN1LgQkv7bHbKJAs5EhWd
nxEt/Hlk3ODg9d3gV8mlsnZwUKT+twpw1aA08XXXTUm6EdGz2OyC/+sOxL9kLX1j
bhd47F18iMjrjld22VkE+rxSH0Ws8HqA7Oxvdq6R2xCOBNyS36D25q5J08FsEhvM
Kar5CKXiNxTKsbhm7xqC5PD48acWabfbqWE8n/Uxy+QARsIvdLGx14HuqCaVvIiv
TDUHKgLKeBRtRytAVunLKmChZwOgzoy8sHJnxDHO2zTlJQNgJXtxmOTAGytfdELS
S8VZCAeHvsXDf+eW2eHcKJfWjwXj9ZtOyh1QRwVTsMo554WgicEFOwE30z9J4nfr
I8iIZjs9OXYhRvHsXyO466JmdXTBQPfYaJqT4i2pLr0cox7IdMakLXogqzu4sEb9
b91fUlV1YvCXoHzXOP0l382gmxDPi7g4Xl7FtKYCNqEeXxzP4padKar9mK5S4fNB
UvupLnKWnyfjqnN9+BojZns7q2WwMgFLFT49ok8MKzWixtlnEjUwzXYuFrOZnk1P
Ti07NEPhmg4NpGaXutIcSkwsKouLgU9xGqndXHt7CMUADTdA43x7VF8vhV929ven
sBxXVsFy6K2ir40zSbofitzmdHxghm+Hl3s=
-----END
CERTIFICATE-----

Found in path(s):
* /ca-certificates-bundle-20250911-r0-tar-gz/etc/ssl/certs/ca-certificates.crt

1.61 busybox 1.37.0-r19

1.61.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source
of bzip2 version 1.0.4. bzip2 source is distributed
under the following conditions (copied verbatim from LICENSE file)
==

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2006 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must
  not claim that you wrote the original software. If you use this

software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must
not be misrepresented
as being the original software.

4. The name of the author may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.
jseward@bzip.org
bzip2/libbzip2 version 1.0.4 of 20 December 2006
--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included
in its entirety, below). Version 2 is the only version of this license which
this version of BusyBox (or modified versions derived from this one) may be
distributed under.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software Foundation's
software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for
a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent
this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an

announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution
of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed
under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you

received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent

infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make
exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED
ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where
 the full notice is found.

  <one line to give the program's name and a brief idea of what it does.>
  Copyright (C) <year> <name of author>

  This program is free software; you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation; either version 2 of the License, or
  (at your option) any later version.

  This program is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with this program; if not, write to the Free Software
  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice
 like this
when it starts in an interactive mode:

  Gnomovision version 69, Copyright (C) year name of author
  Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
  This is free software, and you are welcome to redistribute it
  under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

 Yoyodyne, Inc., hereby disclaims all copyright interest in the program
 `Gnomovision' (which makes passes at compilers) written by James Hacker.

 <signature of Ty Coon>, 1 April 1989
 Ty Coon, President of

Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

1.62 shadow 4.17.3-r0

1.62.1 Available under license :

SPDX-License-Identifier: BSD-3-Clause

All files under this project either

1. fall under the BSD 3 clause license (by default).

2. carry an SPDX header declaring what license applies.

or

3. list a full custom license

This software is originally

* Copyright (c) 1989 - 1994, Julianne Frances Haugh

* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the copyright holders or contributors may not be used to
*   endorse or promote products derived from this software without
*   specific
 prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
* PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
License

All new source code committed to the shadow project is assumed to be made
available under the [BSD-3-Clause](../../COPYING) license unless the submitter
specifies another license at that time. The shadow maintainers reserve the
right to refuse a submission if the license is deemed incompatible with the
goals of the project.

Note: old code may be made available under another license, check the
license tag for each file to get additional information.

1.63 libncursesw 6.5_p20250503-r0

1.63.1 Available under license :

Upstream source https://invisible-island.net/ncurses/Ada95.html

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2023,2024 by Thomas E. Dickey
Copyright: 1998-2016,2017 Free Software Foundation, Inc.
License: X11

Files: aclocal.m4 package
Copyright: 2010-2023,2024 by Thomas E. Dickey
License: X11

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, distribute with modifications, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included

in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS

OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright
holders shall not be used in advertising or otherwise to promote the
sale, use or other dealings in this Software without prior written
authorization.

Files: install-sh
Copyright: 1994 X Consortium
License: X11

Permission is hereby granted, free of charge, to any person obtaining a
copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNEC-
TION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except
as contained in this notice, the name of the X Consortium shall not
be used in advertising or otherwise to promote the sale, use or other deal-
ings in this Software without prior written authorization from the X Consor-
tium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent
`make' implicit rules from creating a file called install from it
when there is no Makefile.

This script is compatible with the BSD install script, but was written

from scratch. It can only install one file at a time, a restriction
shared with many OS's install programs.

On Debian systems, the complete text of the GNU General
Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8
Upstream source https://invisible-island.net/ncurses/ncurses-examples.html

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2023,2024 Thomas E. Dickey
Copyright: 1998-2016,2017 Free Software Foundation, Inc.
License: X11

Files: aclocal.m4 package
Copyright: 2003-2023,2024 by Thomas E. Dickey
License: X11

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, distribute with modifications, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright
holders shall not be used in advertising or otherwise to promote the
sale, use or other dealings in this Software without prior written
authorization.

Files: install-sh
Copyright: 1994 X Consortium

License: X11

Permission is hereby granted, free of charge, to any person obtaining
a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNEC-
TION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not
be used in advertising or otherwise to promote the sale, use or other deal-
ings in this Software without prior written authorization from the X Consor-
tium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent
`make' implicit rules from creating a file called install from it
when there is no Makefile.

This script is compatible with the BSD install script, but was written
from scratch. It can only install one file at a time, a restriction
shared with many OS's install programs.

On Debian systems, the complete text of the GNU General
Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8
Upstream source https://invisible-island.net/ncurses/ncurses.html
This package is used for testing builds of ncurses.

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2023,2024 by Thomas E. Dickey
Copyright: 1998-2016,2017 Free Software Foundation, Inc.

License: X11

Files: aclocal.m4 package
Copyright: 1996-2023,2024 by Thomas E. Dickey
License: X11

Files: doc/html/NCURSES-Programming-HOWTO.html
Copyright: 2001 by Pradeep Padala
License: X11

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, distribute with modifications, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software
is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright
holders shall not be used in advertising or otherwise to promote the
sale, use or other dealings in this Software without prior written
authorization.

Files:
install-sh
Copyright: 1994 X Consortium
License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF
CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNEC-
TION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not
be used in advertising or otherwise to promote the sale, use or other deal-
ings in this Software without prior written authorization from the X Consor-
tium.

Files: progs/tset.c ncurses/tinfo/read_termcap.c
Copyright: 1980,1991,1992,1993 The Regents of the University of California.
License: BSD

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list
of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

-- vile: txtmode file-encoding=utf-8

Copyright 2018-2023,2024 Thomas E. Dickey
Copyright 1998-2017,2018 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, distribute with modifications, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER
 IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright
holders shall not be used in advertising or otherwise to promote the
sale, use or other dealings in this Software without prior written
authorization.

-- vile:txtmode fc=72
-- $Id: COPYING,v 1.13 2024/01/05 21:13:17 tom Exp $

1.64 ncurses 6.5_p20250503-r0

1.64.1 Available under license :

```
# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
 cdefs.h
 queue.h
 tree.h
```

```
"
builddir="$srcdir"
options="!check" # just headers

package() {
 mkdir -p "$pkgdir"
 install -Dm644 -t "$pkgdir"/usr/include/sys \
  cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178  cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750  queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3  tree.h
"
# Automatically generated by apkbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
  INSTALLDIRS=vendor \
  NO_PACKLIST=1 \
  NO_PERLLOCAL=1
 make
}

check() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 make test
```

```
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
  BSD-Resource-1.2911.tar.gz
"
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright"                  # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
 gpep517 build-wheel \
  --wheel-dir .dist \
  --output-fd 3 3>&1 >&2
}

package() {
 gpep517 install-wheel --destdir "$pkgdir" \
  .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
  flake8-copyright-0.2.4.tar.gz
"
```

```
#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef  __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__ ((__nothrow__))
# define __NTH(fct) __attribute__ ((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct)     fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
 default_prepare

 cargo fetch --target="$CTARGET" --locked
}
```

```
build() {
 cargo auditable build --release --frozen
}

check() {
 cargo test --frozen
}

package() {
 install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"
/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
 AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_
```

```
#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y)      \
 ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) ||   \
  (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

/*
* This file defines data structures for different types of trees:
* splay trees and red-black trees.
*
* A splay tree is a self-organizing
 data structure.  Every operation
* on the tree causes a splay to happen.  The splay moves the requested
* node to the root of the tree and partly rebalances it.
*
* This has the benefit that request locality causes faster lookups as
* the requested nodes move to the top of the tree.  On the other hand,
* every lookup causes memory writes.
*
* The Balance Theorem bounds the total access time for m operations
* and n inserts on an initially empty tree as O((m + n)lg n).  The
* amortized cost for a sequence of m accesses to a splay tree is O(lg n);
*
* A red-black tree is a binary search tree with the node color as an
* extra attribute.  It fulfills a set of conditions:
* - every search path from the root to a leaf consists of the
*   same number of black nodes,
* - each red node (except for the root) has a black parent,
* - each leaf node is black.
*
* Every operation on a red-black tree is bounded as O(lg n).
* The maximum height of a red-black tree is 2lg (n+1).
 */

#define SPLAY_HEAD(name, type)      \
struct name {       \
 struct type *sph_root; /* root of the tree */   \
}

#define SPLAY_INITIALIZER(root)      \
```

```
 { NULL }

#define SPLAY_INIT(root) do {      \
 (root)->sph_root = NULL;     \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type)      \
struct {        \
 struct type *spe_left; /* left element */   \
 struct type *spe_right; /* right element */   \
}

#define SPLAY_LEFT(elm, field)  (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field)  (elm)->field.spe_right
#define SPLAY_ROOT(head)  (head)->sph_root
#define SPLAY_EMPTY(head)  (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do {   \
 SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
 SPLAY_RIGHT(tmp, field) = (head)->sph_root;   \
 (head)->sph_root = tmp;      \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do {   \
 SPLAY_RIGHT((head)->sph_root,
 field) = SPLAY_LEFT(tmp, field); \
 SPLAY_LEFT(tmp, field) = (head)->sph_root;   \
 (head)->sph_root = tmp;      \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do {    \
 SPLAY_LEFT(tmp, field) = (head)->sph_root;   \
 tmp = (head)->sph_root;      \
 (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do {    \
 SPLAY_RIGHT(tmp, field) = (head)->sph_root;   \
 tmp = (head)->sph_root;      \
 (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do {  \
 SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
 SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field);\
 SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
 SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
```

```
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
 SPLAY_PROTOTYPE(name, type, field, cmp)    \
void name##_SPLAY(struct name *, struct type *);   \
void name##_SPLAY_MINMAX(struct name *, int);    \
struct type *name##_SPLAY_INSERT(struct name *, struct type *);  \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *);  \
        \
/* Finds the node with the same key as elm */    \
static __inline struct type *      \
name##_SPLAY_FIND(struct name *head, struct type *elm)   \
{        \
 if (SPLAY_EMPTY(head))      \
  return(NULL);      \
 name##_SPLAY(head, elm);     \
 if ((cmp)(elm, (head)->sph_root) == 0)    \
  return (head->sph_root);    \
 return (NULL);       \
}        \
        \
static __inline _sys_tree_h_unused struct type *   \
name##_SPLAY_NEXT(struct name *head, struct type *elm)   \
{        \
 name##_SPLAY(head, elm);     \
 if (SPLAY_RIGHT(elm, field) != NULL) {    \
  elm = SPLAY_RIGHT(elm, field);    \
  while (SPLAY_LEFT(elm, field) != NULL) {  \
   elm = SPLAY_LEFT(elm, field);   \
  }       \
 }
 else       \
  elm = NULL;      \
 return (elm);       \
}        \
        \
static _sys_tree_h_unused __inline struct type *   \
name##_SPLAY_MIN_MAX(struct name *head, int val)   \
{        \
 name##_SPLAY_MINMAX(head, val);     \
       return (SPLAY_ROOT(head));     \
}

/* Main splay operation.
* Moves node close to the key of elm to top
*/
```

```
#define SPLAY_GENERATE(name, type, field, cmp)    \
struct type *       \
name##_SPLAY_INSERT(struct name *head, struct type *elm)  \
{       \
  if (SPLAY_EMPTY(head)) {      \
    SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
  } else {       \
    int __comp;       \
    name##_SPLAY(head, elm);     \
    __comp = (cmp)(elm, (head)->sph_root);   \
    if(__comp < 0) {      \
     SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field);\
     SPLAY_RIGHT(elm, field) = (head)->sph_root;  \
     SPLAY_LEFT((head)->sph_root, field) = NULL;  \
    } else if (__comp > 0) {     \
     SPLAY_RIGHT(elm,
 field) = SPLAY_RIGHT((head)->sph_root, field);\
     SPLAY_LEFT(elm, field) = (head)->sph_root;  \
     SPLAY_RIGHT((head)->sph_root, field) = NULL; \
    } else       \
     return ((head)->sph_root);    \
  }       \
  (head)->sph_root = (elm);      \
  return (NULL);       \
}       \
       \
struct type *       \
name##_SPLAY_REMOVE(struct name *head, struct type *elm)  \
{       \
 struct type *__tmp;      \
 if (SPLAY_EMPTY(head))      \
  return (NULL);      \
 name##_SPLAY(head, elm);     \
 if ((cmp)(elm, (head)->sph_root) == 0) {   \
  if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
   (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
  } else {      \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
   name##_SPLAY(head, elm);   \
   SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
  }       \
  return (elm);      \
 }       \
 return (NULL);       \
}       \
       \
void        \
```

```
name##_SPLAY(struct
 name *head, struct type *elm)   \
{        \
 struct type __node, *__left, *__right, *__tmp;   \
 int __comp;       \
\
 SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
 __left = __right = &__node;     \
\
 while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) {  \
  if (__comp < 0) {     \
   __tmp = SPLAY_LEFT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if ((cmp)(elm, __tmp) < 0){   \
    SPLAY_ROTATE_RIGHT(head, __tmp, field); \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
     break;    \
   }      \
   SPLAY_LINKLEFT(head, __right, field);  \
  } else if (__comp > 0) {    \
   __tmp = SPLAY_RIGHT((head)->sph_root, field); \
   if (__tmp == NULL)    \
    break;     \
   if ((cmp)(elm, __tmp) > 0){   \
    SPLAY_ROTATE_LEFT(head, __tmp, field); \
    if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
     break;    \
   }      \
   SPLAY_LINKRIGHT(head, __left, field);  \
  }       \
 }       \
 SPLAY_ASSEMBLE(head,
 &__node, __left, __right, field);  \
}        \
        \
/* Splay with either the minimum or the maximum element   \
* Used to find minimum or maximum element in tree.   \
*/        \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{        \
 struct type __node, *__left, *__right, *__tmp;   \
\
 SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
 __left = __right = &__node;     \
\
 while (1) {       \
  if (__comp < 0) {     \
```

```
  __tmp = SPLAY_LEFT((head)->sph_root, field); \
  if (__tmp == NULL)    \
   break;     \
  if (__comp < 0){    \
   SPLAY_ROTATE_RIGHT(head, __tmp, field); \
   if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
    break;    \
  }      \
  SPLAY_LINKLEFT(head, __right, field);  \
 } else if (__comp > 0) {    \
  __tmp = SPLAY_RIGHT((head)->sph_root, field); \
  if (__tmp == NULL)    \
   break;     \
  if (__comp > 0) {    \
   SPLAY_ROTATE_LEFT(head, __tmp, field); \
   if (SPLAY_RIGHT((head)->sph_root,
 field) == NULL)\
    break;    \
  }      \
  SPLAY_LINKRIGHT(head, __left, field);  \
 }       \
}       \
 SPLAY_ASSEMBLE(head, &__node, __left, __right, field);  \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y)  name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y)  name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x)  (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x)  (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head)     \
 for ((x) = SPLAY_MIN(name, head);    \
      (x) != NULL;      \
      (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type)      \
struct name {        \
 struct type *rbh_root; /* root of the tree */   \
}
```

```
#define RB_INITIALIZER(root)	\
 {
 NULL }

#define RB_INIT(root) do {	\
 (root)->rbh_root = NULL;	\
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED  1
#define RB_ENTRY(type)	\
struct {	\
 struct type *rbe_left;  /* left element */  \
 struct type *rbe_right;  /* right element */  \
 struct type *rbe_parent; /* parent element */  \
 int rbe_color;   /* node color */  \
}

#define RB_LEFT(elm, field)  (elm)->field.rbe_left
#define RB_RIGHT(elm, field)  (elm)->field.rbe_right
#define RB_PARENT(elm, field)  (elm)->field.rbe_parent
#define RB_COLOR(elm, field)  (elm)->field.rbe_color
#define RB_ROOT(head)   (head)->rbh_root
#define RB_EMPTY(head)   (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do {	\
 RB_PARENT(elm, field) = parent;	\
 RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL;  \
 RB_COLOR(elm, field) = RB_RED;	\
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do {	\
 RB_COLOR(black, field) = RB_BLACK;	\
 RB_COLOR(red, field)
 = RB_RED;	\
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do {	\
 (tmp) = RB_RIGHT(elm, field);	\
 if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
  RB_PARENT(RB_LEFT(tmp, field), field) = (elm);  \
 }	\
 RB_AUGMENT(elm);	\
 if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
```

```
 if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
  RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
 else      \
  RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
 } else       \
 (head)->rbh_root = (tmp);    \
 RB_LEFT(tmp, field) = (elm);     \
 RB_PARENT(elm, field) = (tmp);     \
 RB_AUGMENT(tmp);      \
 if ((RB_PARENT(tmp, field)))     \
  RB_AUGMENT(RB_PARENT(tmp, field));   \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do {   \
 (tmp) = RB_LEFT(elm, field);     \
 if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field))
 != NULL) { \
  RB_PARENT(RB_RIGHT(tmp, field), field) = (elm);  \
 }       \
 RB_AUGMENT(elm);      \
 if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
  if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
   RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
  else      \
   RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
 } else       \
  (head)->rbh_root = (tmp);    \
 RB_RIGHT(tmp, field) = (elm);     \
 RB_PARENT(elm, field) = (tmp);     \
 RB_AUGMENT(tmp);      \
 if ((RB_PARENT(tmp, field)))     \
  RB_AUGMENT(RB_PARENT(tmp, field));   \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp)    \
 RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp)   \
 RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr)  \
attr void name##_RB_INSERT_COLOR(struct name *, struct
 type *);  \
attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
attr struct type *name##_RB_INSERT(struct name *, struct type *); \
attr struct type *name##_RB_FIND(struct name *, struct type *);  \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *);   \
```

```
attr struct type *name##_RB_PREV(struct type *);   \
attr struct type *name##_RB_MINMAX(struct name *, int);   \
        \

/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp)    \
 RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp)   \
 RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr)  \
attr void       \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm)  \
{       \
 struct type
 *parent, *gparent, *tmp;    \
 while ((parent = RB_PARENT(elm, field)) != NULL &&  \
     RB_COLOR(parent, field) == RB_RED) {   \
  gparent = RB_PARENT(parent, field);   \
  if (parent == RB_LEFT(gparent, field)) {  \
   tmp = RB_RIGHT(gparent, field);   \
   if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent;    \
    continue;    \
   }      \
   if (RB_RIGHT(parent, field) == elm) {  \
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    tmp = parent;    \
    parent = elm;    \
    elm = tmp;    \
   }      \
   RB_SET_BLACKRED(parent, gparent, field); \
   RB_ROTATE_RIGHT(head, gparent, tmp, field); \
  } else {      \
   tmp = RB_LEFT(gparent, field);   \
   if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent;    \
    continue;    \
   }      \
   if (RB_LEFT(parent, field) == elm)
  {  \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent;    \
```

```
    parent = elm;    \
    elm = tmp;    \
   }      \
   RB_SET_BLACKRED(parent, gparent, field); \
   RB_ROTATE_LEFT(head, gparent, tmp, field); \
  }       \
 }        \
 RB_COLOR(head->rbh_root, field) = RB_BLACK;   \
}         \
        \
attr void        \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{         \
 struct type *tmp;      \
 while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
     elm != RB_ROOT(head)) {     \
  if (RB_LEFT(parent, field) == elm) {   \
   tmp = RB_RIGHT(parent, field);   \
   if (RB_COLOR(tmp, field) == RB_RED) {  \
    RB_SET_BLACKRED(tmp, parent, field); \
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    tmp = RB_RIGHT(parent, field);  \
   }      \
   if ((RB_LEFT(tmp, field) == NULL ||  \
       RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
       (RB_RIGHT(tmp, field) == NULL ||  \
       RB_COLOR(RB_RIGHT(tmp,
 field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED;  \
    elm = parent;    \
    parent = RB_PARENT(elm, field);  \
   } else {     \
    if (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
     struct type *oleft;  \
     if ((oleft = RB_LEFT(tmp, field)) \
         != NULL)   \
      RB_COLOR(oleft, field) = RB_BLACK;\
     RB_COLOR(tmp, field) = RB_RED; \
     RB_ROTATE_RIGHT(head, tmp, oleft, field);\
     tmp = RB_RIGHT(parent, field); \
    }     \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_RIGHT(tmp, field))  \
     RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    elm = RB_ROOT(head);   \
```

```
    break;     \
   }      \
  } else {      \
   tmp = RB_LEFT(parent, field);   \
   if (RB_COLOR(tmp, field) == RB_RED) {  \
    RB_SET_BLACKRED(tmp, parent, field); \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp
 = RB_LEFT(parent, field);  \
   }      \
   if ((RB_LEFT(tmp, field) == NULL ||  \
       RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
       (RB_RIGHT(tmp, field) == NULL ||  \
       RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED;  \
    elm = parent;    \
    parent = RB_PARENT(elm, field);  \
   } else {     \
    if (RB_LEFT(tmp, field) == NULL ||\
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
     struct type *oright;  \
     if ((oright = RB_RIGHT(tmp, field)) \
         != NULL)   \
      RB_COLOR(oright, field) = RB_BLACK;\
     RB_COLOR(tmp, field) = RB_RED; \
     RB_ROTATE_LEFT(head, tmp, oright, field);\
     tmp = RB_LEFT(parent, field); \
    }     \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_LEFT(tmp, field))  \
     RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    elm = RB_ROOT(head);   \
    break;     \
   }      \
  }       \
 }       \
 if
 (elm)      \
  RB_COLOR(elm, field) = RB_BLACK;   \
}        \
        \
attr struct type *       \
name##_RB_REMOVE(struct name *head, struct type *elm)   \
{        \
 struct type *child, *parent, *old = elm;   \
 int color;       \
```

```
if (RB_LEFT(elm, field) == NULL)    \
 child = RB_RIGHT(elm, field);    \
else if (RB_RIGHT(elm, field) == NULL)    \
 child = RB_LEFT(elm, field);    \
else {        \
 struct type *left;     \
 elm = RB_RIGHT(elm, field);    \
 while ((left = RB_LEFT(elm, field)) != NULL)  \
  elm = left;     \
 child = RB_RIGHT(elm, field);    \
 parent = RB_PARENT(elm, field);    \
 color = RB_COLOR(elm, field);    \
 if (child)      \
  RB_PARENT(child, field) = parent;  \
 if (parent) {      \
  if (RB_LEFT(parent, field) == elm)  \
   RB_LEFT(parent, field) = child;  \
  else      \
   RB_RIGHT(parent, field) = child; \
  RB_AUGMENT(parent);    \
 } else       \
  RB_ROOT(head)
= child;    \
 if (RB_PARENT(elm, field) == old)   \
  parent = elm;     \
 (elm)->field = (old)->field;    \
 if (RB_PARENT(old, field)) {    \
  if (RB_LEFT(RB_PARENT(old, field), field) == old)\
   RB_LEFT(RB_PARENT(old, field), field) = elm;\
  else      \
   RB_RIGHT(RB_PARENT(old, field), field) = elm;\
  RB_AUGMENT(RB_PARENT(old, field));  \
 } else       \
  RB_ROOT(head) = elm;    \
 RB_PARENT(RB_LEFT(old, field), field) = elm;  \
 if (RB_RIGHT(old, field))    \
  RB_PARENT(RB_RIGHT(old, field), field) = elm; \
 if (parent) {      \
  left = parent;     \
  do {      \
   RB_AUGMENT(left);   \
  } while ((left = RB_PARENT(left, field)) != NULL); \
 }       \
 goto color;      \
}        \
parent = RB_PARENT(elm, field);     \
color = RB_COLOR(elm, field);     \
if (child)       \
```

```
 RB_PARENT(child, field) = parent;   \
 if (parent) {       \
  if (RB_LEFT(parent, field) == elm)   \
   RB_LEFT(parent, field) = child;   \
  else       \
   RB_RIGHT(parent,
 field) = child;  \
  RB_AUGMENT(parent);     \
 } else       \
  RB_ROOT(head) = child;     \
color:        \
 if (color == RB_BLACK)      \
  name##_RB_REMOVE_COLOR(head, parent, child);  \
 return (old);       \
}        \
        \
/* Inserts a node into the RB tree */     \
attr struct type *       \
name##_RB_INSERT(struct name *head, struct type *elm)   \
{        \
 struct type *tmp;      \
 struct type *parent = NULL;     \
 int comp = 0;       \
 tmp = RB_ROOT(head);      \
 while (tmp) {       \
  parent = tmp;      \
  comp = (cmp)(elm, parent);    \
  if (comp < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else if (comp > 0)     \
   tmp = RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }        \
 RB_SET(elm, parent, field);     \
 if (parent != NULL) {      \
  if (comp < 0)      \
   RB_LEFT(parent, field) = elm;   \
  else       \
   RB_RIGHT(parent, field) = elm;   \
  RB_AUGMENT(parent);     \
 } else       \
  RB_ROOT(head) = elm;     \
 name##_RB_INSERT_COLOR(head,
 elm);    \
 return (NULL);       \
}        \
        \
```

```
/* Finds the node with the same key as elm */    \
attr struct type *       \
name##_RB_FIND(struct name *head, struct type *elm)   \
{        \
 struct type *tmp = RB_ROOT(head);    \
 int comp;       \
 while (tmp) {       \
  comp = cmp(elm, tmp);     \
  if (comp < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else if (comp > 0)     \
   tmp = RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }       \
 return (NULL);       \
}        \
        \
/* Finds the first node greater than or equal to the search key */ \
attr struct type *       \
name##_RB_NFIND(struct name *head, struct type *elm)   \
{        \
 struct type *tmp = RB_ROOT(head);    \
 struct type *res = NULL;     \
 int comp;       \
 while (tmp) {       \
  comp = cmp(elm, tmp);     \
  if (comp < 0) {      \
   res = tmp;     \
   tmp = RB_LEFT(tmp, field);   \
  }       \
  else if (comp > 0)     \
   tmp =
 RB_RIGHT(tmp, field);   \
  else       \
   return (tmp);     \
 }       \
 return (res);       \
}        \
        \
/* ARGSUSED */       \
attr struct type *       \
name##_RB_NEXT(struct type *elm)     \
{        \
 if (RB_RIGHT(elm, field)) {     \
  elm = RB_RIGHT(elm, field);    \
  while (RB_LEFT(elm, field))    \
   elm = RB_LEFT(elm, field);   \
```

```
 } else {       \
  if (RB_PARENT(elm, field) &&    \
      (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
   elm = RB_PARENT(elm, field);   \
  else {       \
   while (RB_PARENT(elm, field) &&   \
       (elm == RB_RIGHT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field);  \
   elm = RB_PARENT(elm, field);   \
  }       \
 }       \
 return (elm);       \
}        \
        \
/* ARGSUSED */       \
attr struct type *       \
name##_RB_PREV(struct type *elm)     \
{        \
 if (RB_LEFT(elm, field)) {     \
  elm = RB_LEFT(elm, field);    \
  while (RB_RIGHT(elm, field))    \
   elm = RB_RIGHT(elm, field);   \
 }
 else {       \
  if (RB_PARENT(elm, field) &&    \
      (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
   elm = RB_PARENT(elm, field);   \
  else {       \
   while (RB_PARENT(elm, field) &&   \
       (elm == RB_LEFT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field);  \
   elm = RB_PARENT(elm, field);   \
  }       \
 }       \
 return (elm);       \
}        \
        \
attr struct type *       \
name##_RB_MINMAX(struct name *head, int val)    \
{        \
 struct type *tmp = RB_ROOT(head);    \
 struct type *parent = NULL;     \
 while (tmp) {       \
  parent = tmp;      \
  if (val < 0)      \
   tmp = RB_LEFT(tmp, field);   \
  else       \
   tmp = RB_RIGHT(tmp, field);   \
```

```
 }       \
 return (parent);      \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
 y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x)  name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x)  name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head)     \
 for ((x) = RB_MIN(name, head);     \
      (x) != NULL;      \
      (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y)     \
 for ((x) = (y);       \
     ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y)    \
 for ((x) = RB_MIN(name, head);     \
     ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head)    \
 for ((x) = RB_MAX(name, head);     \
      (x) != NULL;      \
      (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y)    \
 for ((x) = (y);       \
     ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
      (x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,
 head, y)   \
 for ((x) = RB_MAX(name, head);     \
     ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
      (x) = (y))
```

#endif /* _SYS_TREE_H_ */
MIT
MIT License

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT
 OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION
 WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl

```
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check"  # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2
 "

builddir="$srcdir"/libart_lgpl-$pkgver

prepare() {
 update_config_sub
 default_prepare
}

build() {
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr
 make
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020  libart_lgpl-2.3.21.tar.bz2"
```

mini_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.65 utmps-libs 0.1.3.1-r0

1.65.1 Available under license :

Copyright (c) 2018-2025 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.66 iputils-ping 20240905-r0

1.66.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether

gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which
gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the
Program does.

1. You may copy and distribute verbatim copies of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to

exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution
and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source
code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the

integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to
decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing
version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU
General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

```
  Gnomovision version 69, Copyright (C) year name of author
  Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
  This is free software, and you are welcome to redistribute
it
  under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

```
 Yoyodyne, Inc., hereby disclaims all copyright interest in the program
 `Gnomovision' (which makes passes at compilers) written by James Hacker.

 <signature of Ty Coon>, 1 April 1989
 Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
 is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.67 github.com/aws/aws-sdk-go-v2/service/internal/presigned-url 1.13.11

1.67.1 Available under license :

GetLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation

```
 spanInitializeEnd
Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}
```

```
type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
```

ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

```
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
```

```
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
  }
 }
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

```
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
```

```
//
//   - Fingerprint
//
//   - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
```

```
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
        "type": "structure",
        "members": {
          "DirectoryId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
            "traits": {
              "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
          },
          "ActiveDirectorySettings":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

           "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
```

```
        },
        "DomainIpv4List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
            }
          }
        },
        "DomainIpv6List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
            }
          }
        },
        "DomainCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
          }
        },
        "DomainNetworkSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
```

```
                "name": "SELF_MANAGED",
                 "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
              },
              {
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Arn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
          },
          "output": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
```

```
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
    }
```

```
},
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

      {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/CreateLicenseServerEndpoint"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
```

```
{
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                  "smithy.api#required": {}
               }
            },
            "LicenseServerSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
               }
            },
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
               }
            }
```

```
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                    "smithy.api#required": {}

  }
                }
            },
            "traits": {
                "smithy.api#input": {}
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
            "type": "structure",
            "members": {
                "LicenseServerEndpoint": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
                    "traits": {
                        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
```

```
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

    },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },

    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
              }
            },
            "Value": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
                "smithy.api#required": {}
              }
```

```
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
            "smithy.api#required": {}
          }
        },
        "Products": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
            "smithy.api#required": {}
          }
        },
        "LastStatusCheckDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                "smithy.api#required": {}
            }
```

```
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {

                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
```

```
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
```

```
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
 },
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
 },
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
```

```
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
                "builtIn": "AWS::Region",
                "required": false,
```

```
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
```

```
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

{
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },

    "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
```

```
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
               "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },

true
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {

"fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
```

```
                    "argv": [
                      {
                        "ref": "PartitionResult"
                      },
                      "supportsFIPS"
                    ]
                  }
                ]
              },

              {
                "fn": "booleanEquals",
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [
                      {
                        "ref": "PartitionResult"
                      },
                      "supportsDualStack"
                    ]
                  }

            ]
              }
            ],
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }

        ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
            "type": "error"
          }
```

```
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [

{
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                    {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
```

```
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"

}
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {

    "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]

     }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
```

```
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                    "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
```

```
"expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",

              "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

   "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
```

```
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
```

```
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,

         "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
```

```
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server.</p>"
            }
        },
        "HealthStatus": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>"
            }
        },
```

```
        "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
            }
        },
        "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
        },
        "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
        },
        "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
        },
        "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
```

```
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }

  }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
```

```
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
             "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
             }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
```

```
criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
             }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",
```

```
      "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
                }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
         "type": "structure",
         "members": {
            "ProductUserSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
```

```
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
            }
        },
        "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
        },
        "SubscriptionEndDate": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
 {
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }
```

```
        },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
```

```
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
          "type": "structure",
          "members": {
            "SecretId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                  "min": 1
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 5,
              "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
          "type": "structure",
          "members": {
            "Endpoint": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
              }
            }
          },
```

```
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
```

```
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
 {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
```

```
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }

         }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StringList": {
          "type": "list",
```

```
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

"min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
```

```
        ],
        "traits": {
          "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
```

```
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

            }
            },
            "TagKeys": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related
```

configuration settings. You can update any combination of settings in a single operation such as the: <ul> Subnets which you want to add to provision VPC endpoints.   Subnets which you want to remove the VPC endpoints from.   Security group ID which permits traffic to the VPC endpoints. ",
                "smithy.api#required": {}
            }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": " The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers. ",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": " The ID of one or more subnets to remove. ",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",

```
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
```

```
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
```

```
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region:
      region,
 ServiceID:     ServiceID,
 OperationName: "GetLicense",
```

```
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
```

```
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
```

```
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string
```

```
// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata
```

```
// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
```

```
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options)) (*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
```

```
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
```

```
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
```

```
//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string
```

```
 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if
err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                 "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of name to filter by.</p>"
                 }
            },
```

```
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
              "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
          "type":
 "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
          },
          "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
```

```
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
           "traits": {
             "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
```

```
            },
            "UsageOperation": {
               "target": "smithy.api#String",

   "traits": {
                  "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
               }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
            },
            "LastUpdatedTime": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
               }
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
          },
          "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
```

```
"operations": [
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
         "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
```

```
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
```

```
                {
                  "conditions": [
                    {
                      "fn": "isSet",
                      "argv": [
                        {
                          "ref": "Endpoint"
                        }
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseFIPS"
                            },
                            true
                          ]
                        }

        ],
                      "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                  "ref": "UseDualStack"
                                },
                                true
                              ]

}
                          ],
                          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                          "type": "error"
                        },
                        {
```

```
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

        "fn": "aws.partition",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ],
                    "assign": "PartitionResult"
                  }
                ],
                "rules": [
                  {
                    "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {

                "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }

],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"

            },
                                "supportsFIPS"
                          ]
                        }
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
```

```
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {

        "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
```

```
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

"fn": "booleanEquals",
                            "argv": [
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true

                        ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
```

```
                "conditions": [],
                "error": "FIPS is enabled
but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true

        ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {

                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [

{
                        "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
```

```
                "type": "error"
              }
            ],
            "type": "tree"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

        }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {

         "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true
```

```
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
          }

    },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {

   "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
```

```
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }

       },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
```

```
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
```

```
      }
    },

  {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

   "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
```

```
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
 <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
```

```
streamline results:</p>\n <ul>\n <li>\n <p>\n <code>AccountID</code>\n
</p>\n </li>\n <li>\n <p>\n <code>AmiID</code>\n </p>\n </li>\n
<li>\n <p>\n <code>DualSubscription</code>\n </p>\n </li>\n
<li>\n <p>\n <code>InstanceID</code>\n </p>\n </li>\n <li>\n
<p>\n <code>InstanceType</code>\n </p>\n </li>\n <li>\n <p>\n
<code>ProductCode</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Region</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Status</code>\n </p>\n </li>\n <li>\n <p>\n
<code>UsageOperation</code>\n </p>\n </li>\n </ul>\n <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n <ul>\n
<li>\n <p>\n <code>contains</code>\n </p>\n </li>\n <li>\n
<p>\n <code>equals</code>\n </p>\n </li>\n <li>\n
<p>\n <code>Notequal</code>\n </p>\n </li>\n </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 16384
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
```

```
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n       <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n        <p>The valid names for this filter are:</p>\n        <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n        </ul>\n        <p>The valid Operators for this filter are:</p>\n        <ul>\n          <li>\n            <p>\n        <code>contains</code>\n          </p>\n        </li>\n        <li>\n          <p>\n        <code>equals</code>\n          </p>\n        </li>\n        <li>\n          <p>\n        <code>Notequal</code>\n          </p>\n        </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

               }
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
         "type": "operation",
         "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
         },
         "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "RegisteredSubscriptionProviders"
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
```

```
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "tags": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "EQUAL",
            "value": "Equal",
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {

"name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },

     {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
```

```
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
               "traits": {
                 "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                 "smithy.api#required": {}
               }
            },
            "SecretArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
              "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've
stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat
Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                 "smithy.api#required": {}
               }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
               }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                 "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
```

```
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
```

```
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
```

```
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
    {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
    resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
```

```
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
```

```
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
```

```
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
{}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
```

```
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
                }
             }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.68 aws-sdk-go-v2-internal-configsources 1.4.11

1.68.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
```

```
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
```

```
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
```

```
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
```

```
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning

Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
```

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32
```

```
 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
```

```
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
```

```
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
 spanInitializeStart
 InterceptExecution
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 spanInitializeEnd
 Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
```

```
        "DomainIpv6List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
            }
          }
        },
        "DomainCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
          }
        },
        "DomainNetworkSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
            "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
        "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }

  },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

          {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
         "type": "structure",
         "members":
 {
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
              }
            },
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
          "type": "union",
          "members": {
            "SecretsManagerCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials needed for user administration in
```

```
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"
```

```
        },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
```

```
    },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
          }
```

```
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
               }
            },
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },

   "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
         "type": "operation",
         "input": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
```

```
identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
                 }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
```

```
    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
```

```
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

               "smithy.api#required": {}
            }
          },
          "Product": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
            }

          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
```

```
            "name": "license-manager-user-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id",
              "content-length",
               "x-api-key",
              "authorization",
              "x-amz-date",
              "x-amz-security-token",
              "Access-Control-Allow-Headers",
              "Access-Control-Allow-Methods",
              "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager User Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {

      "Region": {
                  "builtIn": "AWS::Region",
                  "required": false,
                  "documentation": "The AWS region used to dispatch the request.",
                  "type": "string"
                },
                "UseDualStack": {
                  "builtIn": "AWS::UseDualStack",
                  "required": true,
                  "default": false,
                  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                  "type": "boolean"
                },
                "UseFIPS": {
```

```
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
              }
            },
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Endpoint"
                      }
                    ]

}
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [
```

```
{
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },

                "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "aws.partition",
                         "argv": [
                          {
                            "ref": "Region"
                          }
```

```
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

true

```
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
```

"fn": "booleanEquals",

```
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },
```

```
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]
                                        }

                                ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }

        ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
```

```
{
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

            {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"

  }
        ],
```

```
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

         }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
```

```
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",

      "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
```

```
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS":
true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS":
```

```
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

    "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,

     "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
```

```
                    "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                     }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
            "type": "structure",
            "members": {
              "ProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                  "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
                }
              },
              "HealthStatus": {
                "target":
```

```
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
         }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
```

```
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
              }
            },
            "LicenseServerEndpointId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
              }
            },
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code> resource for the RDS license server.</p>"
              }
            },
            "LicenseServerEndpointProvisioningStatus": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
              }
            },
            "LicenseServers": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
              "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
              }
            },
            "CreationTime": {
              "target": "smithy.api#Timestamp",

    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",

  "value": "DELETED",
            "documentation": ""
          }
        ]
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
       "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
          }
```

```
  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
```

```
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
```

```
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
```

```
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints </p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
```

```
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
           "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
```

```
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

  "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }

          },
            "Settings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
               "traits": {
                  "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
```

```
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
```

```
        "type": "union",

      "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
```

```
                    "min": 1
                },
                "smithy.api#required": {}
            }

    },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
```

```
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
```

```
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
            },
            "IdentityProvider":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) of the product user. "
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": " The domain name of the Active Directory that contains the user for whom to stop the product subscription. "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": " Metadata that describes the start product subscription operation. ",
                    "smithy.api#required": {}
                }

    }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {

```
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {

        "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
```

```
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
          "smithy.api#http": {

   "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns ...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns, c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
```

```
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
```

```
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
```

```
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
```

```
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]
```

```
permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
     - name: Run Scan code on pr ref
       run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
```

```
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
```

```
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
```

```
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
```

```
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
```

```
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
```

```
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
```

```
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
```

```
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

```
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}
```

```
type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
```

```
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
```

```
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
                 "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
               }
            },
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
```

```
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

       "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
                }
            },
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
                }
            },
            "LastSuccessfulDataRetrievalTime": {
                "target": "smithy.api#String",
                "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/subscription/GetServiceSettings"

          },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {}
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
              }
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
```

subscriptions-usage-operation.html\">Usage\n operation values in the License Manager User Guide. "
}
},
"ProductCode": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
"traits": {
"smithy.api#documentation": " The product code for the instance. For more information, see Usage\n operation values in the License Manager User Guide . "
}
},
"LastUpdatedTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The time in which the last discovery updated the instance details. "
}
},
"SubscriptionName": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the license subscription that the instance uses. "
}
},
"OsVersion": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The operating system software version that runs on your instance. "
}
},
"SubscriptionProviderCreateTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses. "
}
},
"SubscriptionProviderUpdateTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider. "
}
},
"DualSubscription": {
"target": "smithy.api#String",

```
            "traits": {
                "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
          },
```

```
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ],
    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
```

```
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
```

```
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": {
                          "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
```

```
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

                                "ref": "UseFIPS"
                                        },
```

```
                true
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }

  ],
                "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"

        },
                            "supportsFIPS"
                        ]
                      }
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [

                         {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                      }
                    ]
```

```
                }
            ],
            "rules": [
                {
                    "conditions": [],

        "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],

                "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        }
    ],
    "type": "tree"
},
{

    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
```

```
                "conditions": [
                  {

            "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true

                ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS is enabled
 but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
```

```
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true

            ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {

                                  "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                              ]
                            }
                          ]
                        }
                      ],
                      "rules": [
                        {
                          "conditions": [],
                          "rules": [

    {
                                "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
```

```
                                        }
                                    ],
                                    "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
],
"type": "tree"
}
]
},
```

```
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

  }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
```

```
                }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
```

```
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
```

```
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
```

```
                },
                {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
                      }
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                },

          {
                    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                    "expect": {
```

```
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",

            "expect": {
                "error": "Invalid Configuration: Missing Region"
              }
            }
          ],
          "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
          "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n          <ul>\n            <li>\n              <p>\n                <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n              <p>\n                <code>AmiID</code>\n              </p>\n            </li>\n
          <li>\n              <p>\n                <code>DualSubscription</code>\n              </p>\n            </li>\n
 <li>\n              <p>\n                <code>InstanceID</code>\n              </p>\n            </li>\n            <li>\n
  <p>\n                <code>InstanceType</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
    <code>ProductCode</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
 <code>Region</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
```

<code>Status</code>\n \n \n \n \n <code>UsageOperation</code>\n \n \n \n For each filter, you can use one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
}
},
"MaxResults": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
"traits": {
"smithy.api#documentation": " The maximum items to return in a request. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
"smithy.api#length": {
"min": 1,
"max": 16384
}
}
}
},
"traits": {
"smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
"type": "structure",
"members": {
"Instances": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
"traits": {
"smithy.api#documentation": " An array that contains instance objects. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
objects. "
}

```
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

              "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
```

```
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
match a certain subscription's name.</p>\n          <p>The valid names for this filter are:</p>\n          <ul>\n
<li>\n              <p>\n                  <code>Subscription</code>\n              </p>\n          </li>\n          </ul>\n
<p>The valid Operators for this filter are:</p>\n          <ul>\n          <li>\n              <p>\n
<code>contains</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
<code>equals</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
        <code>Notequal</code>\n              </p>\n          </li>\n          </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
```

```
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",
```

```
    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
        "smithy.api#http": {
          "method": "GET",
          "uri": "/tags/{resourceArn}"
        },
        "smithy.api#readonly": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
```

```
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
             },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
```

```
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
              },
              {

        "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
               "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
               "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

             "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
               "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
```

```
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
                }
            },
             "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
                }
            },
```

```
            "SubscriptionProviderStatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
               }
            },
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {

                  "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
         }
      },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
```

```
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
```

```
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
 {
        "type": "list",
        "member": {
```

```
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
```

```
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
          "type": "map",
          "key":
{
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
```

```
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

  "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tagKeys": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
```

```
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
 {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required":
{}
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowUpdate": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
          }
        }

      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }
```

```
    },
            "Status": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
               }
            },
            "StatusMessage": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
               "traits": {
                  "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
               }
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
             }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.69 libcrypto3 3.5.4-r0

1.69.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under
the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

a) the GNU General Public License as published by the Free
Software Foundation; either version 1, or (at your option) any
later version, or
b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which

contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the
Program a copy of this General Public License
along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive
use
in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the

recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution
conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY
SERVICING,
REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 1, or (at your option)
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free
Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will

not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard

names,
and clearly documenting the differences in manual pages (or
equivalent), together with instructions on where to get the Standard
Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this
Package. You may charge any fee you choose for support of this Package. You
may not charge a fee for this Package itself. However, you may distribute this
Package in aggregate with other (possibly commercial) programs as part of a
larger (possibly commercial) software distribution provided that you do not
advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output
from the programs of this Package do not automatically fall under the copyright
of this Package, but belong to whomever generated them, and may be sold
commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package
shall not
be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote
products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.70 alpine-baselayout-data 3.7.0-r0

1.70.1 Available under license :

```
# Automatically generated by apkbuild-cpan, template 1
# Contributor: Francesco Colista <francesco.colista@gmail.com>
# Maintainer: Francesco Colista <francesco.colista@gmail.com>
pkgname=perl-software-license
_pkgreal=Software-License
pkgver=0.103004
pkgrel=0
pkgdesc="packages that provide templated software licenses"
url="http://search.cpan.org/dist/Software-License/"
arch="noarch"
license="GPL PerlArtistic"
cpandepends="perl-text-template perl-data-section perl-sub-install"
cpanmakedepends="    "
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/R/RJ/RJBS/$_pkgreal-$pkgver.tar.gz"

_builddir="$srcdir/$_pkgreal-$pkgver"
```

```
prepare() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 make && make test
}

package() {
 cd "$_builddir"
 make
 DESTDIR="$pkgdir" install || return 1
 find "$pkgdir" \( -name perllocal.pod -o -name .packlist \) -delete
}

md5sums="0c5fd53d93818566234578b27d667b35  Software-License-0.103004.tar.gz"
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=5
pkgdesc="A library for high-performance 2D graphics"
url="http://www.levien.com/libart/"
arch="all"
license="LGPL"
depends=
makedepends=
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2
 "

_builddir="$srcdir"/libart_lgpl-$pkgver

prepare() {
 cd "$_builddir"
 update_config_sub || return 1
}
build() {
 cd "$_builddir"
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr \
  || return 1
```

```
 make || return 1
}

package() {
 cd "$_builddir"
 make DESTDIR="$pkgdir" install || return 1
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020  libart_lgpl-2.3.21.tar.bz2"
```

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.
Do not use or load this software and any associated materials
(collectively, the "Software") until you have carefully read the
following terms and conditions. By loading or using the Software, you
agree to the terms of this Agreement. If you do not wish to so agree,
do not install or use the Software.

1. COPYRIGHT NOTICE
Some or all of this work - Copyright 1999-2011, Intel Corp. All
rights reserved.

2. LICENSE

2.1. This is your license from Intel Corp. under its intellectual
property rights. You may have additional license terms from the party
that provided you this software, covering your right to use that
party's intellectual property rights.

2.2. Intel grants, free of charge, to any person ("Licensee")
obtaining a copy of the source code appearing in this file ("Covered
Code") an irrevocable, perpetual, worldwide license under Intel's
copyrights in the base code distributed originally by Intel ("Original
Intel
 Code") to copy, make derivatives, distribute, use and display
any portion of the Covered Code in any form, with the right to
sublicense such rights; and

2.3. Intel grants Licensee a non-exclusive and non-transferable patent
license (with the right to sublicense), under only those claims of
Intel patents that are infringed by the Original Intel Code, to make,
use, sell, offer to sell, and import the Covered Code and derivative
works thereof solely to the minimum extent necessary to exercise the
above copyright license, and in no event shall the patent license
extend to any additions to or modifications of the Original Intel
Code. No other license or right is granted directly or by implication,
estoppel or otherwise; The above copyright and patent license is
granted only if the following conditions are met:

3. CONDITIONS

3.1. Redistribution of Source with Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or
modification with rights to further distribute source must include the above Copyright Notice, the above License, this list of Conditions, and the following Disclaimer and Export Compliance provision. In addition, Licensee must cause all Covered Code to which Licensee contributes to contain a file documenting the changes Licensee made to create that Covered Code and the date of any change. Licensee must include in that file the documentation of any changes made by any predecessor Licensee. Licensee must include a prominent statement that the modification is derived, directly or indirectly, from Original Intel Code.

3.2. Redistribution of Source with no Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification without rights to further distribute source must include the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with distribution. In addition,
Licensee may not authorize further sublicense of source of any portion of the Covered Code, and must include terms to the effect that the license from Licensee to its licensee is limited to the intellectual property embodied in the software Licensee provides to its licensee, and not to intellectual property embodied in modifications its licensee may make.

3.3. Redistribution of Executable. Redistribution in executable form of any substantial portion of the Covered Code or modification must reproduce the above Copyright Notice, and the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with the distribution.

3.4. Intel retains all right, title, and interest in and to the Original Intel Code.

3.5. Neither the name Intel nor any other trademark owned or controlled by Intel shall be used in advertising or otherwise to promote the sale, use or other dealings in products derived from or relating to the Covered
Code without prior written authorization from Intel.

4. DISCLAIMER AND EXPORT COMPLIANCE

4.1. INTEL MAKES NO WARRANTY OF ANY KIND REGARDING ANY SOFTWARE

PROVIDED HERE. ANY SOFTWARE ORIGINATING FROM INTEL OR DERIVED FROM INTEL SOFTWARE IS PROVIDED "AS IS," AND INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS. INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

4.2. IN NO EVENT SHALL INTEL HAVE ANY LIABILITY TO LICENSEE, ITS LICENSEES OR ANY OTHER THIRD PARTY, FOR ANY LOST PROFITS, LOST DATA, LOSS OF USE OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, AND IRRESPECTIVE OF WHETHER INTEL HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS
SHALL APPLY NOTWITHSTANDING THE
FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

4.3. Licensee shall not export, either directly or indirectly, any of this software or system incorporating such software without first obtaining any required license or other approval from the U. S. Department of Commerce or any other agency or department of the United States Government. In the event Licensee exports any such software from the United States or re-exports any such software from a foreign destination, Licensee shall ensure that the distribution and export/re-export of the software is in compliance with all laws, regulations, orders, or other restrictions of the U.S. Export Administration Regulations. Licensee agrees that neither it nor any of its subsidiaries will export/re-export any technical data, process, software, or service, directly or indirectly, to any country for which the United States government or any agency thereof requires an export license, other governmental
approval, or letter of assurance, without
first obtaining such license, approval or letter.
INTEL SOFTWARE LICENSE AGREEMENT

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.
Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

LICENSES: Please Note:
- If you are a network administrator, the "Site License" below shall apply to you.
- If you are an end user, the "Single User License" shall apply to you.
- If you are an original equipment manufacturer (OEM), the "OEM License" shall apply to you.

SITE LICENSE. You may copy the Software onto your organization's computers for your organization's use, and you may make a reasonable number of back-up copies of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component
products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

SINGLE USER LICENSE. You may copy the Software onto a single computer for your personal, noncommercial use, and you may make one back-up copy of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part
of
the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

OEM LICENSE: You may reproduce and distribute the Software only as an integral part of or incorporated in Your product or as a standalone Software maintenance update for existing end users of Your products, excluding any other standalone products, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this
Agreement, and you agree to
prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may only distribute the Software to your customers pursuant to a

written license agreement. Such license agreement may be a "break-the-seal" license agreement. At a minimum such license shall safeguard Intel's ownership rights to the Software.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

NO OTHER RIGHTS. No rights or licenses are granted by Intel to You, expressly or by implication, with respect to any proprietary information or patent, copyright, mask work, trademark, trade secret, or other intellectual property right owned or controlled by Intel, except as expressly provided in this Agreement.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected
by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights, trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS
PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement
at any time
if you violate its terms. Upon termination, you will immediately destroy the
Software or return all copies of the Software to Intel.

APPLICABLE LAWS. Claims arising under this Agreement shall be governed by the
laws of California, excluding its principles of conflict of laws and the
United Nations Convention on Contracts for the Sale of Goods. You may not
export the Software in violation of applicable export laws and regulations.
Intel is not obligated under any other agreements unless they are in writing
and signed by an authorized representative of Intel.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED
RIGHTS." Use, duplication, or disclosure by the Government is subject to
restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its
successor. Use of the Software by the Government constitutes acknowledgment
of Intel's proprietary rights therein. Contractor or Manufacturer is Intel
2200 Mission College Blvd., Santa Clara, CA
95052.
/*
* Copyright (c) 1983, 1988 Regents of the University of California.
* All rights reserved.
*
* Redistribution and use in source and binary forms are permitted
* provided that the above copyright notice and this paragraph are
* duplicated in all such forms and that any documentation,
* advertising materials, and other materials related to such
* distribution and use acknowledge that the software was developed
* by the University of California, Berkeley. The name of the
* University may not be used to endorse or promote products derived
* from this software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
* WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
*/
Automatically generated by apkbuild-cpan, template 1
Contributor: Valery Kartel <valery.kartel@gmail.com>
Maintainer: Valery Kartel <valery.kartel@gmail.com>
pkgname=perl-bsd-resource
_pkgreal=BSD-Resource
pkgver=1.2909
pkgrel=2
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="http://search.cpan.org/dist/BSD-Resource/"
arch="all"
license="GPL PerlArtistic"
cpandepends=""
cpanmakedepends="   "

```
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/J/JH/JHI/$_pkgreal-$pkgver.tar.gz"

_builddir="$srcdir/$_pkgreal-$pkgver"

prepare() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 make && make test || return 1
}

package() {
 cd "$_builddir"
 make DESTDIR="$pkgdir" install || return 1
 find "$pkgdir"
 \( -name perllocal.pod -o -name .packlist \) -delete
}

md5sums="3c6dd2fa953088aa87e263f40818d012  BSD-Resource-1.2909.tar.gz"
sha256sums="c47d460e2bd00d8d049f6ff4a0f1f61957e289e8de45af5fc79e851e2e855e06  BSD-Resource-
1.2909.tar.gz"
sha512sums="5f489eaac5089e82081b7cf51d62638750e443751b49415407354da96b17771819b5e6e85529faa41ddd
85743363750157943c7f7700276cce2bd40a429fa430  BSD-Resource-1.2909.tar.gz"
SQLite Copyright
SQLite is in the
Public Domain
```

All of the deliverable code in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Portions of the documentation and some code used as part of the build process might fall under other licenses. The details here are unclear. We do not worry about the licensing of the documentation and build code so much because none
 of these things are part of the core deliverable SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.
Obtaining An Explicit License To Use SQLite

Even though SQLite is in the public domain and does not require a license, some users want to obtain a license anyway. Some reasons for obtaining a license include:
You are using SQLite in a jurisdiction that does not recognize the public domain.
You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain.
You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.
Your legal department tells
you that you have to purchase a license.

If you feel like you really have to purchase a license for SQLite, Hwaci, the company that employs the architect and principal developers of SQLite, will sell you one.
Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, all new contributors to the SQLite code base are asked to dedicate their contributions to the public domain. If you want to send a patch or enhancement for possible inclusion in the SQLite source tree, please accompany the patch with the following statement:
The author or authors of this code dedicate any and all copyright interest in this code to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this code under copyright law.

We are not able to accept patches or changes to SQLite that are not accompanied
by a statement such as the above. In addition, if you make changes or enhancements as an employee, then a simple statement such as the above is insufficient. You must also send by surface mail a copyright release signed by a company officer. A signed original of the copyright release should be mailed to:
Hwaci
6200 Maple Cove Lane
Charlotte, NC 28269
USA

A template copyright release is available in PDF or HTML. You can use this release to make future changes.

see http://www.sqlite.org/copyright.html
Copyright (c) 2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software source and associated documentation files (the "Materials"),
to deal in the Materials without restriction, including without limitation
the rights to use, copy, modify, compile, merge, publish, distribute,
sublicense, and/or sell copies of the Materials, and to permit persons to
whom the Materials are furnished to do so, subject the following terms and
conditions:

All modifications to the Materials used to create a binary that is
distributed to third parties shall be provided to Khronos with an
unrestricted license to use for the purposes of implementing bug fixes and
enhancements to the Materials;

If the binary is used as part of an OpenCL(TM) implementation, whether binary
is distributed together with or separately to that implementation, then
recipient must become an OpenCL Adopter and follow the published OpenCL
conformance process for that implementation,
details at:
http://www.khronos.org/conformance/;

The above copyright notice, the OpenCL trademark license, and this permission
notice shall be included in all copies or substantial portions of the
Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

OpenCL is a trademark of Apple Inc. used under license by Khronos.
Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.0
pkgrel=1
pkgdesc="Extension for flake8 which checks for copyrights"
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="flake8"
makedepends="python3-dev"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

build() {
 cd "$builddir"
 python3 setup.py build || return 1
}

package() {
 cd "$builddir"

```
 python3 setup.py install --prefix=/usr --root="$pkgdir" || return 1
}

md5sums="c6348990c6d74933bf5d5f6a55a14bdf  flake8-copyright-0.2.0.tar.gz"
sha256sums="aeef26eb4d5223c9cd5b101e68175fcef6d2b353bf36da688fdde62fccfe2b73  flake8-copyright-
0.2.0.tar.gz"
sha512sums="bab1bfaeed3a525ac3c7da12b1f2ec338c204cc69b72d05cd39560e528faff5d586c3b1be9570f4a977a7e
2417b586e543bfba12abce256b2023c8c4993eeb90
  flake8-copyright-0.2.0.tar.gz"
mini_sendmail - accept email on behalf of real sendmail

Copyright  1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
```

1.71 github.com/aws/aws-sdk-go-v2/internal/v4a 1.4.11

1.71.1 Available under license :

```
GetLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
```

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
```

ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde

```
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
```

```
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
```

```
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
```

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent

```
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
```

```
 params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
```

```
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
```

```
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
```

```
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
```

```
//
//   - ProductSKU
//
//   - Fingerprint
//
//   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
```

```
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
          "type": "structure",
          "members": {
            "DirectoryId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
              "traits": {
                "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
              }
            },
            "ActiveDirectorySettings":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
              }
            },
            "ActiveDirectoryType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
              "traits": {
                "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
              }
            },
            "IsSharedActiveDirectory": {
              "target": "smithy.api#Boolean",
              "traits": {

           "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
          "type": "structure",
          "members": {
            "DomainName": {
              "target": "smithy.api#String",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
        },
        "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
        },
        "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
```

```
            "smithy.api#enum": [
               {
                  "name": "SELF_MANAGED",
                   "value": "SELF_MANAGED",
                  "documentation": "SELF_MANAGED type of Active Directory"
               },
               {
                  "name": "AWS_MANAGED",
                  "value": "AWS_MANAGED",
                  "documentation": "AWS_MANAGED type of Active Directory"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
         },
         "output": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
 {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The identity provider for the user.</p>",

             "smithy.api#required": {}
               }
            },
```

```
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                 "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
            "smithy.api#error": "server"
        }
```

```
},
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },

      {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/CreateLicenseServerEndpoint"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
```

```
{
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                  "smithy.api#required": {}
               }
            },
            "LicenseServerSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                  "smithy.api#required": {}
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
               }
            },
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
               }
            }
```

```
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/DeleteLicenseServerEndpoint"

          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "LicenseServerEndpoint": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
              "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
```

```
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

    },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },

    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
              }
            },
            "Value": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
                "smithy.api#required": {}
              }
```

```
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
      "type": "structure",
      "members": {
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
            "smithy.api#required": {}
          }
        },
        "Products": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
          }
        },
        "LastStatusCheckDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                "smithy.api#required": {}
            }
```

```
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {

                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
```

```
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
```

```
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
 },
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
 },
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
   "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
```

```
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
```

```
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
```

```
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

{
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },

    "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
```

```
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "rules": [
                      {
                        "conditions": [
                          {

"fn": "booleanEquals",
                            "argv": [
                              true,
                              {
                                "fn": "getAttr",
```

```
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            },

            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }

            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
```

```
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [

{
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                    {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
```

```
                    "type": "tree"
                  },
                  {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"

}
                ],
                "type": "tree"
              },
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {

    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsDualStack"
                            ]

     }
                          ]
                        }
                      ],
                      "rules": [
                        {
                          "conditions": [],
                          "endpoint": {
```

```
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }

                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                    "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
```

```
"expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",

              "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

   "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
```

```
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
```

```
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,

         "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
```

```
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server.</p>"
            }
        },
        "HealthStatus": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>"
            }
        },
```

```
        "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
            }
        },
        "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
        },
        "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
        },
        "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
        },
        "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
```

```
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }

  }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
```

```
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
             "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
             }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
```

```
criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
             }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",
```

```
      "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
                }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
         "type": "structure",
         "members": {
            "ProductUserSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
```

```
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                "smithy.api#required": {}
            }
        },
        "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
        },
        "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
        },
        "SubscriptionEndDate": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
                }
```

```
        },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
```

```
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
          "type": "structure",
          "members": {
            "SecretId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                  "min": 1
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
          "type": "string",
          "traits": {
            "smithy.api#length": {
              "min": 5,
              "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
          "type": "structure",
          "members": {
            "Endpoint": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
              }
            }
          },
```

```
        "traits": {
          "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

       "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
```

```
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
 {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
```

```
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
              }
            },
            "ProductUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
              }

         }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StringList": {
          "type": "list",
```

```
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

"min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
```

```
        ],
        "traits": {
          "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
```

```
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

            }
            },
            "TagKeys": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related
```

configuration settings. You can update any combination of settings in a single operation such as the: <ul> Subnets which you want to add to provision VPC endpoints.   Subnets which you want to remove the VPC endpoints from.   Security group ID which permits traffic to the VPC endpoints. ",
"smithy.api#required": {}
}
}
},
"traits": {
"smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
"type": "structure",
"members": {
"IdentityProviderSummary": {
"target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
"traits": {
"smithy.api#required": {}
}
}
},
"traits": {
"smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
"type": "structure",
"members": {
"AddSubnets": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
"traits": {
"smithy.api#documentation": " The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers. ",
"smithy.api#required": {}
}
},
"RemoveSubnets": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
"traits": {
"smithy.api#documentation": " The ID of one or more subnets to remove. ",
"smithy.api#required": {}
}
},
"SecurityGroupId": {
"target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",

```
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
```

```
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
```

```
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
```

```
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region:
      region,
 ServiceID:     ServiceID,
 OperationName: "GetLicense",
```

```
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
```

```
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
```

```
Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string
```

```
// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata
```

```
// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
```

```
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options)) (*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
```

```
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
```

```
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
```

```
//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err !=
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string
```

```
 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
         "type": "structure",
         "members": {
            "SubscriptionProviderArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
         "type": "structure",
         "members": {},
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
         "type": "structure",
         "members": {
            "Name": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The type of name to filter by.</p>"
               }
            },
```

```
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
              "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
          "type":
 "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
          },
          "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

         "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
                }
            },
```

```
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
           "traits": {
             "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
```

```
            },
            "UsageOperation": {
               "target": "smithy.api#String",

   "traits": {
                  "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
               }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
            },
            "LastUpdatedTime": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
               }
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
               }
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
          },
          "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
```

```
"operations": [
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
         "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
```

```
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
```

```
                {
                  "conditions": [
                    {
                      "fn": "isSet",
                      "argv": [
                        {
                          "ref": "Endpoint"
                        }
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseFIPS"
                            },
                            true
                          ]
                        }

        ],
                      "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                {
                                  "ref": "UseDualStack"
                                },
                                true
                              ]

}
                          ],
                          "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                          "type": "error"
                        },
                        {
```

```
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"

        }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

        "fn": "aws.partition",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ],
                    "assign": "PartitionResult"
                  }
                ],
                "rules": [
                  {
                    "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {

                "ref": "UseFIPS"
                    },
                    true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }

    ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"

            },
                                "supportsFIPS"
                            ]
                          }
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
```

```
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

    "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],

                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                        "type": "error"
                    }
                ],
                "type": "tree"
            },
            {

        "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
```

```
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

                "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true

                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
```

```
                "conditions": [],
                "error": "FIPS is enabled
but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true

        ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {

                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [

{
                        "conditions": [],
```

```
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
```

```
                "type": "error"
            }
          ],
          "type": "tree"
        }
      ]
    },
    "smithy.rules#endpointTests": {
      "testCases": [
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

      }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

       "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
```

```
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
          }

    },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {

  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
```

```
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
          }

       },
        "params": {
          "Region": "us-gov-east-1",
          "UseFIPS": true,
          "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
```

```
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
```

```
      }
    },

  {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

   "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
```

```
                }
            ],
            "version": "1.0"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
 <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
```

```
streamline results:</p>\n <ul>\n <li>\n <p>\n <code>AccountID</code>\n
</p>\n </li>\n <li>\n <p>\n <code>AmiID</code>\n </p>\n </li>\n
<li>\n <p>\n <code>DualSubscription</code>\n </p>\n </li>\n
<li>\n <p>\n <code>InstanceID</code>\n </p>\n </li>\n <li>\n
<p>\n <code>InstanceType</code>\n </p>\n </li>\n <li>\n <p>\n
<code>ProductCode</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Region</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Status</code>\n </p>\n </li>\n <li>\n <p>\n
<code>UsageOperation</code>\n </p>\n </li>\n </ul>\n <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n <ul>\n
<li>\n <p>\n <code>contains</code>\n </p>\n </li>\n <li>\n
<p>\n <code>equals</code>\n </p>\n </li>\n <li>\n
<p>\n <code>Notequal</code>\n </p>\n </li>\n </ul>"
          }
        },
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 16384
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
```

```
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n       <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

               }
            }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
         "type": "operation",
         "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
         },
         "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/ListRegisteredSubscriptionProviders"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "RegisteredSubscriptionProviders"
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

   "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
```

```
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
    },
    "traits": {
      "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
         },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {

"name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },

     {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
```

```
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
               "traits": {
                 "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                 "smithy.api#required": {}
               }
            },
            "SecretArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
              "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've
stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat
Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                 "smithy.api#required": {}
               }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
               }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                 "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
```

```
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
```

```
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
```

```
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
```

```
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
    {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
    resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
```

```
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
```

```
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
```

```
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
{}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
```

```
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
            }

    },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
            }
         }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
      }
    }
  }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.72 aws-sdk-go-v2-service-internal-s3shared 1.19.11

1.72.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
```

```
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
```

```
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
```

```
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
```

```
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning

Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
```

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32
```

```
 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
```

```
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
 return err
 }
 if err = addRetry(stack, options); err != nil {
 return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
 err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
```

```
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
```

```
        "DomainIpv6List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
            }
          }
        },
        "DomainCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
          }
        },
        "DomainNetworkSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
            "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
        "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
```

```
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
             "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
               "smithy.api#required": {}
             }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
          "smithy.api#error": "server"
        }

 },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

          {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
         "type": "structure",
         "members":
 {
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
              }
            },
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code> resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
          "type": "union",
          "members": {
            "SecretsManagerCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials needed for user administration in
```

```
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"
```

```
        },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
```

```
    },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
          }
```

```
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
               }
            },
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },

   "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
         "type": "operation",
         "input": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
```

```
identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
                 }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
```

```
    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
```

```
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

               "smithy.api#required": {}
            }
          },
          "Product": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
            }

          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
```

```
            "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                 "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {

    "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
```

```
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
              }
            },
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Endpoint"
                      }
                    ]

}
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [
```

```
{
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },

              "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "aws.partition",
                         "argv": [
                          {
                            "ref": "Region"
                          }
```

```
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

true

```
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
```

"fn": "booleanEquals",

```
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                      ]
                    }

                ]
                }
              ],
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }

        ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
              "type": "error"
            }
          ],
          "type": "tree"
        },
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
```

```
{
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                    {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"

  }
                ],
```

```
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

         }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
```

```
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",

      "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
```

```
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {

"endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
```

```
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

    "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,

     "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
```

```
                "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
          },
          "HealthStatus": {
            "target":
```

```
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
```

```
            "StatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
               }
            },
            "LicenseServerEndpointId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
               }
            },
            "LicenseServerEndpointArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
               }
            },
            "LicenseServerEndpointProvisioningStatus": {
               "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
               "traits": {
                  "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
               }
            },
            "LicenseServers": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
               "traits": {
                  "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
               }
            },
            "CreationTime": {
               "target": "smithy.api#Timestamp",

    "traits": {
                  "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
         }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",

  "value": "DELETED",
            "documentation": ""
          }
        ]
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
       "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
          }
```

```
  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
```

```
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
```

```
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
```

```
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints </p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
```

```
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
           "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
```

```
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }

          },
            "Settings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
               "traits": {
                  "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
```

```
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
```

```
        "type": "union",

      "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
```

```
                    "min": 1
                },
                "smithy.api#required": {}
            }

    },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
```

```
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
```

```
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
            },
            "IdentityProvider":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) of the product user. "
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": " The domain name of the Active Directory that contains the user for whom to stop the product subscription. "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": " Metadata that describes the start product subscription operation. ",
                    "smithy.api#required": {}
                }

    }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {

```
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {

        "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
```

```
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
          "smithy.api#http": {

    "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
```

```
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
```

```
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
```

```
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
```

```
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]
```

```
permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
     - name: Run Scan code on pr ref
       run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
```

```
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
```

```
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
```

```
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
```

```
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
```

```
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
```

```
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
```

```
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
```

```
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}
```

```
type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
```

```
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
```

```
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
                 "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
               }
            },
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
```

```
            "uri": "/subscription/GetServiceSettings"

          },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {}
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
              }
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
```

```
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
                }
            },
            "ProductCode": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
                "traits": {
                    "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
                }
            },
            "LastUpdatedTime": {
                "target": "smithy.api#String",
                 "traits": {
                    "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
                }
            },
            "SubscriptionName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
                }
            },
            "OsVersion": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
                }
            },
            "SubscriptionProviderCreateTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
                }
            },
            "SubscriptionProviderUpdateTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
                }
            },
            "DualSubscription": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
          },
```

```
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
       "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
```

```
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
```

```
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": {
                          "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
```

```
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

                                "ref": "UseFIPS"
                                        },
```

```
                true
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }

  ],
                "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"

        },
                            "supportsFIPS"
                        ]
                      }
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [

                         {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                      }
                    ]
```

```
                }
            ],
            "rules": [
                {
                    "conditions": [],

        "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],

                "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        }
    ],
    "type": "tree"
},
{

    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
```

```
                "conditions": [
                  {

            "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true

                ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

  "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS is enabled
but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
```

```
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true

            ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {

                                  "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                              ]
                            }
                          ]
                        }
                      ],
                      "rules": [
                        {
                          "conditions": [],
                          "rules": [

    {
                                "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
```

```
                                        }
                                    ],
                                    "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
],
"type": "tree"
}
]
},
```

```
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

  }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
```

```
                }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
```

```
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
```

```
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
                },
                {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
                      }
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                },

          {
                    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                    "expect": {
```

```
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",

            "expect": {
                "error": "Invalid Configuration: Missing Region"
              }
            }
          ],
          "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
          "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n            </li>\n
 <li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>InstanceType</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
 <code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
```

<code>Status</code>\n \n \n \n \n <code>UsageOperation</code>\n \n \n \n For each filter, you can use one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
}
},
"MaxResults": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
"traits": {
"smithy.api#documentation": " The maximum items to return in a request. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
"smithy.api#length": {
"min": 1,
"max": 16384
}
}
}
},
"traits": {
"smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
"type": "structure",
"members": {
"Instances": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
"traits": {
"smithy.api#documentation": " An array that contains instance objects. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
objects. "
}

```
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

              "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
```

```
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
match a certain subscription's name.</p>\n          <p>The valid names for this filter are:</p>\n          <ul>\n
<li>\n              <p>\n                  <code>Subscription</code>\n              </p>\n          </li>\n          </ul>\n
<p>The valid Operators for this filter are:</p>\n          <ul>\n          <li>\n              <p>\n
<code>contains</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
<code>equals</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
        <code>Notequal</code>\n              </p>\n          </li>\n          </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
```

```
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",
```

```
    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
        "smithy.api#http": {
          "method": "GET",
          "uri": "/tags/{resourceArn}"
        },
        "smithy.api#readonly": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
```

```
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
             },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
```

```
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
              },
              {

        "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
               "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
               "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
               "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
```

```
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
                }
            },
             "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
                }
            },
```

```
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
                }
            },
            "LastSuccessfulDataRetrievalTime": {
                "target": "smithy.api#String",
                "traits": {

                    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
```

```
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
```

```
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
 {
        "type": "list",
        "member": {
```

```
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
```

```
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
          "type": "map",
          "key":
{
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
```

```
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

  "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tagKeys": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
```

```
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
 {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required":
{}
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowUpdate": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
          }
        }

      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }
```

```
    },
            "Status": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
               }
            },
            "StatusMessage": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
               "traits": {
                  "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
               }
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
             }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.73 iputils-tracepath 20240905-r0

1.73.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE

POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which
gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1

above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause
any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work
are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable

source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution
and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source
code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in

certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing
version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY

YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU
General Public License as published by
the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate

parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

 Yoyodyne, Inc., hereby disclaims all copyright interest in the program
 `Gnomovision' (which makes passes at compilers) written by James Hacker.

 <signature of Ty Coon>, 1 April 1989
 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
 is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.74 iputils 20240905-r0

1.74.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which

gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the
Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution
and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source
code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot

impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing
version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU
General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.75 alpine-baselayout 3.7.0-r0

1.75.1 Available under license :

```
# Automatically generated by apkbuild-cpan, template 1
# Contributor: Francesco Colista <francesco.colista@gmail.com>
# Maintainer: Francesco Colista <francesco.colista@gmail.com>
pkgname=perl-software-license
_pkgreal=Software-License
pkgver=0.103004
pkgrel=0
pkgdesc="packages that provide templated software licenses"
url="http://search.cpan.org/dist/Software-License/"
arch="noarch"
license="GPL PerlArtistic"
cpandepends="perl-text-template perl-data-section perl-sub-install"
cpanmakedepends="    "
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/R/RJ/RJBS/$_pkgreal-$pkgver.tar.gz"
```

```
_builddir="$srcdir/$_pkgreal-$pkgver"

prepare() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 make && make test
}

package() {
 cd "$_builddir"
 make
 DESTDIR="$pkgdir" install || return 1
 find "$pkgdir" \( -name perllocal.pod -o -name .packlist \) -delete
}

md5sums="0c5fd53d93818566234578b27d667b35  Software-License-0.103004.tar.gz"
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=5
pkgdesc="A library for high-performance 2D graphics"
url="http://www.levien.com/libart/"
arch="all"
license="LGPL"
depends=
makedepends=
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2
 "

_builddir="$srcdir"/libart_lgpl-$pkgver

prepare() {
 cd "$_builddir"
 update_config_sub || return 1
}
build() {
 cd "$_builddir"
 ./configure \
  --build=$CBUILD \
  --host=$CHOST \
```

```
  --prefix=/usr \
  || return 1
 make || return 1
}

package() {
 cd "$_builddir"
 make DESTDIR="$pkgdir" install || return 1
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020  libart_lgpl-2.3.21.tar.bz2"
```

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.
Do not use or load this software and any associated materials
(collectively, the "Software") until you have carefully read the
following terms and conditions. By loading or using the Software, you
agree to the terms of this Agreement. If you do not wish to so agree,
do not install or use the Software.

1. COPYRIGHT NOTICE
Some or all of this work - Copyright 1999-2011, Intel Corp. All
rights reserved.

2. LICENSE

2.1. This is your license from Intel Corp. under its intellectual
property rights. You may have additional license terms from the party
that provided you this software, covering your right to use that
party's intellectual property rights.

2.2. Intel grants, free of charge, to any person ("Licensee")
obtaining a copy of the source code appearing in this file ("Covered
Code") an irrevocable, perpetual, worldwide license under Intel's
copyrights in the base code distributed originally by Intel ("Original
Intel
 Code") to copy, make derivatives, distribute, use and display
any portion of the Covered Code in any form, with the right to
sublicense such rights; and

2.3. Intel grants Licensee a non-exclusive and non-transferable patent
license (with the right to sublicense), under only those claims of
Intel patents that are infringed by the Original Intel Code, to make,
use, sell, offer to sell, and import the Covered Code and derivative
works thereof solely to the minimum extent necessary to exercise the
above copyright license, and in no event shall the patent license
extend to any additions to or modifications of the Original Intel
Code. No other license or right is granted directly or by implication,
estoppel or otherwise; The above copyright and patent license is

granted only if the following conditions are met:

3. CONDITIONS

3.1. Redistribution of Source with Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or
modification with rights to further distribute source must include the above Copyright Notice, the above License, this list of Conditions, and the following Disclaimer and Export Compliance provision. In addition, Licensee must cause all Covered Code to which Licensee contributes to contain a file documenting the changes Licensee made to create that Covered Code and the date of any change. Licensee must include in that file the documentation of any changes made by any predecessor Licensee. Licensee must include a prominent statement that the modification is derived, directly or indirectly, from Original Intel Code.

3.2. Redistribution of Source with no Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification without rights to further distribute source must include the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with distribution. In addition,
Licensee may not authorize further sublicense of source of any portion of the Covered Code, and must include terms to the effect that the license from Licensee to its licensee is limited to the intellectual property embodied in the software Licensee provides to its licensee, and not to intellectual property embodied in modifications its licensee may make.

3.3. Redistribution of Executable. Redistribution in executable form of any substantial portion of the Covered Code or modification must reproduce the above Copyright Notice, and the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with the distribution.

3.4. Intel retains all right, title, and interest in and to the Original Intel Code.

3.5. Neither the name Intel nor any other trademark owned or controlled by Intel shall be used in advertising or otherwise to promote the sale, use or other dealings in products derived from or relating to the Covered
Code without prior written authorization from Intel.

4. DISCLAIMER AND EXPORT COMPLIANCE

4.1. INTEL MAKES NO WARRANTY OF ANY KIND REGARDING ANY SOFTWARE PROVIDED HERE. ANY SOFTWARE ORIGINATING FROM INTEL OR DERIVED FROM INTEL SOFTWARE IS PROVIDED "AS IS," AND INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS. INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

4.2. IN NO EVENT SHALL INTEL HAVE ANY LIABILITY TO LICENSEE, ITS LICENSEES OR ANY OTHER THIRD PARTY, FOR ANY LOST PROFITS, LOST DATA, LOSS OF USE OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, AND IRRESPECTIVE OF WHETHER INTEL HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS
SHALL APPLY NOTWITHSTANDING THE
FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

4.3. Licensee shall not export, either directly or indirectly, any of
this software or system incorporating such software without first
obtaining any required license or other approval from the U. S.
Department of Commerce or any other agency or department of the United
States Government. In the event Licensee exports any such software
from the United States or re-exports any such software from a foreign
destination, Licensee shall ensure that the distribution and
export/re-export of the software is in compliance with all laws,
regulations, orders, or other restrictions of the U.S. Export
Administration Regulations. Licensee agrees that neither it nor any of
its subsidiaries will export/re-export any technical data, process,
software, or service, directly or indirectly, to any country for which
the United States government or any agency thereof requires an export
license, other governmental
approval, or letter of assurance, without
first obtaining such license, approval or letter.
INTEL SOFTWARE LICENSE AGREEMENT

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.
Do not use or load this software and any associated materials (collectively,
the "Software") until you have carefully read the following terms and
conditions. By loading or using the Software, you agree to the terms of this
Agreement. If you do not wish to so agree, do not install or use the Software.

LICENSES: Please Note:
- If you are a network administrator, the "Site License" below shall
apply to you.
- If you are an end user, the "Single User License" shall apply to you.
- If you are an original equipment manufacturer (OEM), the "OEM License"

shall apply to you.

SITE LICENSE. You may copy the Software onto your organization's computers for your organization's use, and you may make a reasonable number of back-up copies of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component
products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

SINGLE USER LICENSE. You may copy the Software onto a single computer for your personal, noncommercial use, and you may make one back-up copy of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part
of
the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

OEM LICENSE: You may reproduce and distribute the Software only as an integral part of or incorporated in Your product or as a standalone Software maintenance update for existing end users of Your products, excluding any other standalone products, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this
Agreement, and you agree to
prevent unauthorized copying of the Software.

3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may only distribute the Software to your customers pursuant to a written license agreement. Such license agreement may be a "break-the-seal" license agreement. At a minimum such license shall safeguard Intel's ownership rights to the Software.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

NO OTHER RIGHTS. No rights or licenses are granted by Intel to You, expressly or by implication, with respect to any proprietary information or patent, copyright, mask work, trademark, trade secret, or other intellectual property right owned or controlled by Intel, except as expressly provided in this Agreement.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected
by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights, trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED
ABOVE, THE SOFTWARE IS
PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE

OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement
at any time
if you violate its terms. Upon termination, you will immediately destroy the
Software or return all copies of the Software to Intel.

APPLICABLE LAWS. Claims arising under this Agreement shall be governed by the
laws of California, excluding its principles of conflict of laws and the
United Nations Convention on Contracts for the Sale of Goods. You may not
export the Software in violation of applicable export laws and regulations.
Intel is not obligated under any other agreements unless they are in writing
and signed by an authorized representative of Intel.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED
RIGHTS." Use, duplication, or disclosure by the Government is subject to
restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its
successor. Use of the Software by the Government constitutes acknowledgment
of Intel's proprietary rights therein. Contractor or Manufacturer is Intel
2200 Mission College Blvd., Santa Clara, CA
95052.
/*
* Copyright (c) 1983, 1988 Regents of the University of California.
* All rights reserved.
*
* Redistribution and use in source and binary forms are permitted
* provided that the above copyright notice and this paragraph are
* duplicated in all such forms and that any documentation,
* advertising materials, and other materials related to such
* distribution and use acknowledge that the software was developed
* by the University of California, Berkeley. The name of the
* University may not be used to endorse or promote products derived
* from this software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
* WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
*/
Automatically generated by apkbuild-cpan, template 1
Contributor: Valery Kartel <valery.kartel@gmail.com>
Maintainer: Valery Kartel <valery.kartel@gmail.com>
pkgname=perl-bsd-resource
_pkgreal=BSD-Resource
pkgver=1.2909
pkgrel=2
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="http://search.cpan.org/dist/BSD-Resource/"
arch="all"
license="GPL PerlArtistic"

```
cpandepends=""
cpanmakedepends="    "
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/J/JH/JHI/$_pkgreal-$pkgver.tar.gz"

_builddir="$srcdir/$_pkgreal-$pkgver"

prepare() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 make && make test || return 1
}

package() {
 cd "$_builddir"
 make DESTDIR="$pkgdir" install || return 1
 find "$pkgdir"
 \( -name perllocal.pod -o -name .packlist \) -delete
}

md5sums="3c6dd2fa953088aa87e263f40818d012  BSD-Resource-1.2909.tar.gz"
sha256sums="c47d460e2bd00d8d049f6ff4a0f1f61957e289e8de45af5fc79e851e2e855e06  BSD-Resource-
1.2909.tar.gz"
sha512sums="5f489eaac5089e82081b7cf51d62638750e443751b49415407354da96b17771819b5e6e85529faa41ddd
85743363750157943c7f7700276cce2bd40a429fa430  BSD-Resource-1.2909.tar.gz"
SQLite Copyright
SQLite is in the
Public Domain
```

All of the deliverable code in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Portions of the documentation and some code used as part of the build process might fall under other licenses. The details here are unclear. We do not worry about the licensing of the documentation and build code so much because none

of these things are part of the core deliverable SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.
Obtaining An Explicit License To Use SQLite

Even though SQLite is in the public domain and does not require a license, some users want to obtain a license anyway. Some reasons for obtaining a license include:
You are using SQLite in a jurisdiction that does not recognize the public domain.
You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain.
You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.
Your legal department tells
you that you have to purchase a license.

If you feel like you really have to purchase a license for SQLite, Hwaci, the company that employs the architect and principal developers of SQLite, will sell you one.
Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, all new contributors to the SQLite code base are asked to dedicate their contributions to the public domain. If you want to send a patch or enhancement for possible inclusion in the SQLite source tree, please accompany the patch with the following statement:
The author or authors of this code dedicate any and all copyright interest in this code to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this code under copyright law.

We are not able to accept patches or changes to SQLite that are not accompanied
by a statement such as the above. In addition, if you make changes or enhancements as an employee, then a simple statement such as the above is insufficient. You must also send by surface mail a copyright release signed by a company officer. A signed original of the copyright release should be mailed to:
Hwaci
6200 Maple Cove Lane
Charlotte, NC 28269
USA

A template copyright release is available in PDF or HTML. You can use this release to make future changes.

see http://www.sqlite.org/copyright.html
Copyright (c) 2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software source and associated documentation files (the "Materials"),
to deal in the Materials without restriction, including without limitation
the rights to use, copy, modify, compile, merge, publish, distribute,
sublicense, and/or sell copies of the Materials, and to permit persons to

whom the Materials are furnished to do so, subject the following terms and
conditions:

All modifications to the Materials used to create a binary that is
distributed to third parties shall be provided to Khronos with an
unrestricted license to use for the purposes of implementing bug fixes and
enhancements to the Materials;

If the binary is used as part of an OpenCL(TM) implementation, whether binary
is distributed together with or separately to that implementation, then
recipient must become an OpenCL Adopter and follow the published OpenCL
conformance process for that implementation,
 details at:
http://www.khronos.org/conformance/;

The above copyright notice, the OpenCL trademark license, and this permission
notice shall be included in all copies or substantial portions of the
Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN
THE MATERIALS.

OpenCL is a trademark of Apple Inc. used under license by Khronos.
Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.0
pkgrel=1
pkgdesc="Extension for flake8 which checks for copyrights"
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="flake8"
makedepends="python3-dev"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/$_pkgname/$_pkgname-$pkgver.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

build() {
 cd "$builddir"
 python3 setup.py build || return 1
}

```
package() {
 cd "$builddir"
 python3 setup.py install --prefix=/usr --root="$pkgdir" || return 1
}

md5sums="c6348990c6d74933bf5d5f6a55a14bdf  flake8-copyright-0.2.0.tar.gz"
sha256sums="aeef26eb4d5223c9cd5b101e68175fcef6d2b353bf36da688fdde62fccfe2b73  flake8-copyright-
0.2.0.tar.gz"
sha512sums="bab1bfaeed3a525ac3c7da12b1f2ec338c204cc69b72d05cd39560e528faff5d586c3b1be9570f4a977a7e
2417b586e543bfba12abce256b2023c8c4993eeb90
  flake8-copyright-0.2.0.tar.gz"
```

mini_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

1.76 alpine-release 3.22.2-r0

1.76.1 Available under license :

```
# MIT License
```

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT
OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
MIT

1.77 baruwa-enterprise-clamd 1.0.1

1.77.1 Available under license :

Mozilla Public License Version 2.0
==================================

1. Definitions

1.1. "Contributor"
means each individual or legal entity that creates, contributes to
the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached
the notice in Exhibit A, the Executable Form of such Source Code
Form, and Modifications of such Source Code Form, in each case
including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of

version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method,

process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark)
Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes
such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes
additional grants as a result of Your choice to
distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the
Source
Code Form of the Covered Software is governed by the terms of this

License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any

such warranty, support, indemnity, or liability obligation is offered by
You alone, and You hereby agree to indemnify every Contributor for any
liability incurred by such Contributor as a result of warranty, support,
indemnity or liability terms You offer. You may include additional
disclaimers of warranty and limitations of liability specific to any
jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this
License with respect to some or all of the Covered Software due to
statute, judicial order, or regulation then You must: (a) comply with
the terms of this License to the maximum
extent possible; and (b)
describe the limitations and the code they affect. Such description must
be placed in a text file included with all distributions of the Covered
Software under this License. Except to the extent prohibited by statute
or regulation, such description must be sufficiently detailed for a
recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically
if You fail to comply with any of its terms. However, if You become
compliant, then the rights granted under this License from a particular
Contributor are reinstated (a) provisionally, unless and until such
Contributor explicitly and finally terminates Your grants, and (b) on an
ongoing basis, if such Contributor fails to notify You of the
non-compliance by some reasonable means prior to 60 days after You have
come back into compliance. Moreover, Your grants from a particular
Contributor are reinstated on an ongoing
basis if such Contributor
notifies You of the non-compliance by some reasonable means, this is the
first time You have received notice of non-compliance with this License
from such Contributor, and You become compliant prior to 30 days after
Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent
infringement claim (excluding declaratory judgment actions,
counter-claims, and cross-claims) alleging that a Contributor Version
directly or indirectly infringes any patent, then the rights granted to
You by any and all Contributors for the Covered Software under Section
2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all
end user license agreements (excluding distributors and resellers) which

have been validly granted by You or Your distributors under this License
prior to termination shall survive termination.

```
************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered
 Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*                                                                      *
*  Under no circumstances and under no legal theory, whether tort      *
*  (including negligence), contract, or otherwise, shall any           *
*  Contributor, or anyone who distributes Covered Software as          *
*  permitted above, be liable to You for any direct, indirect,         *
*  special, incidental, or consequential damages of any character      *
*  including, without limitation,
 damages for lost profits, loss of    *
*  goodwill, work stoppage, computer failure or malfunction, or any    *
*  and all other commercial damages or losses, even if such party      *
*  shall have been informed of the possibility of such damages. This   *
*  limitation of liability shall not apply to liability for death or   *
*  personal injury resulting from such party's negligence to the       *
*  extent applicable law prohibits such limitation. Some               *
*  jurisdictions do not allow the exclusion or limitation of           *
*  incidental or consequential damages, so this exclusion and          *
*  limitation may not apply to You.                                    *
*                                                                      *
************************************************************************
```

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place
of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2.
Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary

Licenses

If You choose to distribute Source Code Form that is Incompatible With
Secondary Licenses under the terms of this version of the License, the
notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source
Code Form is subject to the terms of the Mozilla Public
License, v. 2.0. If a copy of the MPL was not distributed with this
file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular
file, then You may include the notice in a location (such as a LICENSE
file in a relevant directory) where a recipient would be likely to look
for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as
defined by the Mozilla Public License, v. 2.0.

1.78 libbsd 0.12.2-r0

1.78.1 Available under license :

Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/

Files:
*
Copyright:
Copyright 2004-2024 Guillem Jover <guillem@hadrons.org>
License: BSD-3-clause

Files:
include/bsd/err.h
include/bsd/stdlib.h
include/bsd/sys/param.h
include/bsd/unistd.h
src/bsd_getopt.c
src/err.c
src/fgetln.c
src/progname.c

Copyright:
Copyright 2005, 2008-2012, 2019 Guillem Jover <guillem@hadrons.org>
Copyright 2005 Hector Garcia Alvarez
Copyright 2005 Aurelien Jarno
Copyright 2006 Robert Millan
Copyright 2018 Facebook, Inc.
License: BSD-3-clause

Files:
include/bsd/netinet/ip_icmp.h
include/bsd/sys/bitstring.h
include/bsd/sys/queue.h
include/bsd/sys/time.h
include/bsd/timeconv.h
include/bsd/vis.h
man/bitstring.3bsd
man/errc.3bsd
man/explicit_bzero.3bsd
man/fgetln.3bsd
man/fgetwln.3bsd
man/fpurge.3bsd
man/funopen.3bsd
man/getbsize.3bsd
man/heapsort.3bsd
man/nlist.3bsd
man/pwcache.3bsd
man/queue.3bsd
man/radixsort.3bsd
man/reallocarray.3bsd
man/reallocf.3bsd
man/setmode.3bsd
man/strmode.3bsd
man/strnstr.3bsd
man/strtoi.3bsd
man/strtou.3bsd
man/unvis.3bsd
man/vis.3bsd
man/wcslcpy.3bsd
src/getbsize.c
src/heapsort.c
src/merge.c
src/nlist.c
src/pwcache.c
src/radixsort.c
src/setmode.c
src/strmode.c
src/strnstr.c
src/strtoi.c

src/strtou.c
src/unvis.c
Copyright:
Copyright 1980, 1982, 1986, 1989-1994
   The Regents of the University of California. All rights reserved.
Copyright 1992 Keith Muller.
Copyright 2001 Mike Barcroft <mike@FreeBSD.org>
.
Some code is derived from software contributed to Berkeley by
the American National Standards Committee X3, on Information
Processing Systems.
.
Some code is derived from software contributed to Berkeley by
Peter McIlroy.
.
Some code is derived from software contributed to Berkeley by
Ronnie Kon at Mindcraft Inc., Kevin Lew and Elmer Yglesias.
.
Some code is derived from software contributed to Berkeley by
Dave Borman at Cray Research, Inc.
.
Some code is
 derived from software contributed to Berkeley by
Paul Vixie.
.
Some code is derived from software contributed to Berkeley by
Chris Torek.
.
Copyright UNIX System Laboratories, Inc.
All or some portions of this file are derived from material licensed
to the University of California by American Telephone and Telegraph
Co. or Unix System Laboratories, Inc. and are reproduced herein with
the permission of UNIX System Laboratories, Inc.
License: BSD-3-clause-Regents

Files:
src/vis.c
Copyright:
Copyright 1989, 1993
   The Regents of the University of California. All rights reserved.
.
Copyright 1999, 2005 The NetBSD Foundation, Inc.
All rights reserved.
License: BSD-3-clause-Regents and BSD-2-clause-NetBSD

Files:
include/bsd/libutil.h
Copyright:

Copyright 1996 Peter Wemm <peter@FreeBSD.org>.
All rights reserved.
Copyright 2002 Networks Associates Technology, Inc.
All rights reserved.
License: BSD-3-clause-author

Files:
man/timeradd.3bsd
Copyright:
Copyright
 2009 Jukka Ruohonen <jruohonen@iki.fi>
Copyright 1999 Kelly Yancey <kbyanc@posi.net>
All rights reserved.
License: BSD-3-clause-John-Birrell
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the author nor the names of any co-contributors
   may be used to endorse or promote products derived from this software
   without specific prior written permission.
.
THIS SOFTWARE IS PROVIDED BY JOHN BIRRELL AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES
 OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Files:
man/setproctitle.3bsd
Copyright:
Copyright 1995 Peter Wemm <peter@FreeBSD.org>
All rights reserved.
License: BSD-5-clause-Peter-Wemm
Redistribution and use in source and binary forms, with or without
modification, is permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice immediately at the beginning of the file, without
 modification,
   this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. This work was done expressly for inclusion into FreeBSD. Other use
   is permitted provided this notation is included.
4. Absolutely no warranty of function or purpose is made by the author
   Peter Wemm.
5. Modifications may be freely made to this file providing the above
   conditions are met.

Files:
include/bsd/stringlist.h
man/arc4random.3bsd
man/fmtcheck.3bsd
man/humanize_number.3bsd
man/stringlist.3bsd
man/timeval.3bsd
src/fmtcheck.c
src/humanize_number.c
src/stringlist.c
src/strtonum.c
Copyright:
Copyright 1994, 1997-2000, 2002, 2008, 2010, 2014
    The NetBSD Foundation, Inc.
Copyright 2013 John-Mark Gurney <jmg@FreeBSD.org>
All rights reserved.
.
Copyright 2014
 The NetBSD Foundation, Inc.
All rights reserved.
.
Some code was derived from software contributed to The NetBSD Foundation
by Taylor R. Campbell.
.
Some code was contributed to The NetBSD Foundation by Allen Briggs.
.
Some code was contributed to The NetBSD Foundation by Luke Mewburn.
.
Some code is derived from software contributed to The NetBSD Foundation
by Jason R. Thorpe of the Numerical Aerospace Simulation Facility,
NASA Ames Research Center, by Luke Mewburn and by Tomas Svensson.
.
Some code is derived from software contributed to The NetBSD Foundation
by Julio M. Merino Vidal, developed as part of Google's Summer of Code

2005 program.
.
Some code is derived from software contributed to The NetBSD Foundation
by Christos Zoulas.
.
Some code is derived from software contributed to The NetBSD Foundation
by Jukka Ruohonen.
License: BSD-2-clause-NetBSD

Files:
include/bsd/sys/endian.h
man/byteorder.3bsd
man/closefrom.3bsd
man/expand_number.3bsd
man/flopen.3bsd
man/getpeereid.3bsd
man/pidfile.3bsd
src/expand_number.c
src/pidfile.c
src/reallocf.c
src/timeconv.c
Copyright:
Copyright 1998, M. Warner Losh <imp@freebsd.org>
All rights reserved.
.
Copyright 2001 Dima Dorfman.
All rights reserved.
.
Copyright 2001 FreeBSD Inc.
All rights reserved.
.
Copyright 2002 Thomas Moestl <tmm@FreeBSD.org>
All rights reserved.
.
Copyright 2002 Mike Barcroft <mike@FreeBSD.org>
All rights reserved.
.
Copyright 2005 Pawel Jakub Dawidek <pjd@FreeBSD.org>
All rights reserved.
.
Copyright 2005 Colin Percival
All rights reserved.
.
Copyright 2007 Eric Anderson <anderson@FreeBSD.org>
Copyright 2007 Pawel Jakub Dawidek <pjd@FreeBSD.org>
All rights reserved.
.
Copyright 2007 Dag-Erling Codan Smrgrav

All rights reserved.
.
Copyright 2009 Advanced Computing Technologies LLC
All rights reserved.
.
Copyright 2011 Guillem Jover <guillem@hadrons.org>
License: BSD-2-clause

Files:
src/flopen.c
Copyright:
Copyright 2007-2009
 Dag-Erling Codan Smrgrav
All rights reserved.
License: BSD-2-clause-verbatim
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer
   in this position and unchanged.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Files:
include/bsd/sys/tree.h
man/fparseln.3bsd
man/tree.3bsd
src/fparseln.c
Copyright:
Copyright 1997 Christos Zoulas.
All rights reserved.
.
Copyright 2002 Niels Provos <provos@citi.umich.edu>

All rights reserved.
License: BSD-2-clause-author

Files:
include/bsd/readpassphrase.h
man/readpassphrase.3bsd
man/strlcpy.3bsd
man/strtonum.3bsd
src/arc4random.c
src/arc4random_linux.h
src/arc4random_uniform.c
src/arc4random_unix.h
src/arc4random_win.h
src/closefrom.c
src/freezero.c
src/getentropy_aix.c
src/getentropy_bsd.c
src/getentropy_hpux.c
src/getentropy_hurd.c
src/getentropy_linux.c
src/getentropy_osx.c
src/getentropy_solaris.c
src/getentropy_win.c
src/readpassphrase.c
src/reallocarray.c
src/recallocarray.c
src/strlcat.c
src/strlcpy.c
test/explicit_bzero.c
test/strtonum.c
Copyright:
Copyright 2004 Ted Unangst and Todd Miller
All rights reserved.
.
Copyright 1996 David Mazieres <dm@uun.org>
Copyright 1998, 2000-2002, 2004-2005, 2007, 2010, 2012-2015
Todd C. Miller <Todd.Miller@courtesan.com>
Copyright 2004 Ted Unangst
Copyright 2004 Otto Moerbeek <otto@drijf.net>
Copyright 2008 Damien Miller <djm@openbsd.org>
Copyright 2008, 2010-2011, 2016-2017 Otto Moerbeek <otto@drijf.net>
Copyright 2013 Markus Friedl <markus@openbsd.org>
Copyright 2014 Bob Beck <beck@obtuse.com>
Copyright 2014 Brent Cook <bcook@openbsd.org>
Copyright 2014 Pawel Jakub Dawidek <pjd@FreeBSD.org>
Copyright 2014 Theo de Raadt <deraadt@openbsd.org>
Copyright 2014 Google Inc.
Copyright 2015 Michael Felt <aixtools@gmail.com>

Copyright 2015, 2022 Guillem Jover <guillem@hadrons.org>
License:
 ISC
Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.
.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:
src/inet_net_pton.c
Copyright:
Copyright 1996 by Internet Software Consortium.
License: ISC-Original
Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice
 and this permission notice appear in all copies.
.
THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS
ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE
CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR
PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

Files:
src/setproctitle.c
Copyright:
Copyright 2010 William Ahern
Copyright 2012 Guillem Jover <guillem@hadrons.org>
License: Expat
Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute,
 sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the
following conditions:

.
The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.
.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE SOFTWARE.

Files:
src/explicit_bzero.c
src/chacha_private.h
Copyright:
None
License: public-domain
Public domain.

Files:
man/mdX.3bsd
Copyright:
None
License: Beerware
"THE BEER-WARE LICENSE" (Revision 42):
<phk@login.dkuug.dk> wrote this
file. As long as you retain this notice you
can do whatever you want with this stuff. If we meet some day, and you think
this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

License: BSD-3-clause-Regents
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.
.
THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

License: BSD-3-clause-author
Redistribution and use in source and binary forms, with or without
modification, is permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions
 in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior written
   permission.
.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

License: BSD-3-clause
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.
.
THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES,

INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-NetBSD
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
.
THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-author
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above
copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

License: BSD-2-clause
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

1.79 stdlib go1.25.0

1.79.1 Available under license :

```
// Copyright 2024 The Go Authors. All rights reserved.
// Use of this source code is governed by a BSD-style
// license that can be found in the LICENSE file.

package copyright

import (
 "bytes"
 "internal/testenv"
 "io"
 "io/fs"
 "os"
 "path/filepath"
```

```
 "testing"
)

var copyright = []byte("Copyright")

var permitted = [][]byte{
 []byte("// Code generated by "),
 []byte("// Code generated from "),
 []byte("// Created by cgo -cdefs"),
 []byte("// DO NOT EDIT\n// generated by:"),
 []byte("// Empty assembly file"),
 []byte("// Generated using cgo"),
 []byte("// Original source:\n//\thttp://www.zorinaq.com/papers/md5-amd64.html"), // public domain crypto/md5
 []byte("// created by cgo -cdefs"),
 []byte("// go run mkasm.go"),
 []byte("// mkerrors"),
 []byte("// mksys"),
 []byte("// run\n// Code generated by"), // cmd/compile/internal/test/constFold_test.go
}

func TestCopyright(t *testing.T) {
 buf := make([]byte, 2048)
 filepath.WalkDir(filepath.Join(testenv.GOROOT(t), "src"), func(path string,
 d fs.DirEntry, err error) error {
  if d.IsDir() && (d.Name() == "testdata" || d.Name() == "vendor") {
   return filepath.SkipDir
  }
  switch filepath.Ext(d.Name()) {
  default:
   return nil
  case ".s", ".go":
   // check
  }

  f, err := os.Open(path)
  if err != nil {
   t.Error(err)
   return nil
  }
  defer f.Close()
  n, err := f.Read(buf)
  if err != nil && err != io.EOF {
   t.Error(err)
   return nil
  }
  b := buf[:n]
  if bytes.Contains(b, copyright) {
   return nil
```

```
  }
  for _, ok := range permitted {
   if bytes.HasPrefix(b, ok) {
    return nil
   }
  }
  t.Errorf("%s: missing copyright notice", path)
  return nil
 })
}
```

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google LLC nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE
 FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The Go source code and supporting files in this directory
are covered by the usual Go license (see ../../../../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto_linux_amd64.syso object file is built
from BoringSSL source code by build/build.sh and is covered
by the BoringSSL license reproduced below and also at
https://boringssl.googlesource.com/boringssl/+/fips-20220613/LICENSE.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL
licensing. Files that are completely new have a Google copyright and an ISC
license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:
https://cla.developers.google.com/clas

Some files from Intel are under yet another license, which is also included
underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the
OpenSSL License and the original SSLeay license apply to the toolkit.
 See below
for the actual license texts. Actually both licenses are BSD-style Open Source
licenses. In case of any license issues related to OpenSSL please contact
openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from
some authors is recorded for use of their work. (This is purely for our own
record keeping.)
 27287199
 27287880
 27287883

 OpenSSL License

/* ==
* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following
 disclaimer in
*   the documentation and/or other materials provided with the
*   distribution.
*
* 3. All advertising materials mentioning features or use of this
*   software must display the following acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*

* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*   endorse or promote products derived from this software without
*   prior written permission. For written permission, please contact
*   openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*   nor may "OpenSSL" appear in their names without prior written
*   permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit
 (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product
 includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions

* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of
 the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are
 not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence

* [including the GNU Public Licence.]
*/

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.
*
* Permission to use, copy, modify, and/or distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
* OF CONTRACT, NEGLIGENCE
 OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

Some files from Intel carry the following license:

Copyright (c) 2012, Intel Corporation
#
All rights reserved.
#
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
#
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
#
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.
#
* Neither the name of the Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.
#
#
THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION
 ""AS IS"" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.
#
https://developers.google.com/open-source/cla/individual
https://developers.google.com/open-source/cla/corporate
#
Names should be added to this file as:
Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyoun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google LLC nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE
 FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY
EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY
AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the
authors and should not be interpreted as representing official policies, either expressed
or implied, of Andrea Leofreddi.

1.80 confluentinc-confluent-kafka-go 2.12.0

1.80.1 Available under license :

LICENSE
---
librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill
2023, Confluent Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished
to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

For src/crc32c.c copied (with modifications) from
http://stackoverflow.com/a/17646775/1821055

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
This software is provided 'as-is',
without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,

including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler
madler@alumni.caltech.edu
*/

LICENSE.fnv1a
---
parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please
do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:
  chongo <Landon Curt Noll> /\oo/\
  http://www.isthe.com/chongo/

Share and Enjoy! :-)

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram https://github.com/codahale/hdrhistogram
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License

(MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE

LICENSE.lz4

---

src/rdxxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library
Copyright (c) 2011-2020, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.murmur2

---

parts of src/rdmurmur2.c: git@github.com:abrandoned/murmur2.git

MurMurHash2 Library

//----------

// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person
obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.nanopb

For
files in src/nanopb :
https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or
implied warranty. In no event will the authors be held liable
for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you
must not claim that you wrote the original software. If you use
this software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may
not be removed or altered from any source
distribution.

LICENSE.opentelemetry

For files in src/opentelemetry: https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes
of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated
by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution
incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE
text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding
such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event
and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent
with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.
==========

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted,
 free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

LICENSE.queue

For
 sys/queue.h:

* Copyright (c) 1991, 1993
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 4. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR
 IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* $FreeBSD$

LICENSE.regexp
---
regexp.c and regexp.h from https://github.com/ccxvii/minilibs sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"
These libraries are in the public domain (or the equivalent where that is not
 possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.
"

LICENSE.snappy

##
LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h
originally retrieved from http://github.com/andikleen/snappy-c
git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219
##

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions
 in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.tinycthread

From https://github.com/tinycthread/tinycthread/README.txt c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from https://github.com/alex85k/wingetopt

```
/*
* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
 ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*-
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice,
 this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
```

* POSSIBILITY OF SUCH DAMAGE.
*/
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You"
(or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations,
or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems
that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants
to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work

or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution
of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,
or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted
against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.81 github.com/aws/aws-sdk-go-v2/service/sts 1.38.9

1.81.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
```

```
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
```

```
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
```

```
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
```

```
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning

Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
  - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
```

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32
```

```
 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
```

```
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
 return err
 }
 if err = addRetry(stack, options); err != nil {
 return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
 err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
```

```
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
 spanInitializeStart
 InterceptExecution
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 spanInitializeEnd
 Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
```

```
        "DomainIpv6List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
            }
          }
        },
        "DomainCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
          }
        },
        "DomainNetworkSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
            "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
        "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
```

```
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
             "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
               "smithy.api#required": {}
             }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
          "smithy.api#error": "server"
        }

 },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

          {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
              }
            },
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code> resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
          "type": "union",
          "members": {
            "SecretsManagerCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials needed for user administration in
```

```
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"
```

```
        },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
```

```
    },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
          }
```

```
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
               }
            },
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },

   "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
         "type": "operation",
         "input": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
```

```
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
                 }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
```

```
    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
```

```
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
            "ActiveDirectoryIdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

                    "smithy.api#required": {}
                }
            },
            "Product": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
            }

          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
```

```
            "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                 "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
            ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {

    "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
```

```
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
              }
            },
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Endpoint"
                      }
                    ]

}
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [
```

```
{
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },

                "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "aws.partition",
                         "argv": [
                          {
                            "ref": "Region"
                          }
```

```
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

true

```
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
```

"fn": "booleanEquals",

```
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },
```

```
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            true,
                                            {
                                                "fn": "getAttr",
                                                "argv": [
                                                    {
                                                        "ref": "PartitionResult"
                                                    },
                                                    "supportsDualStack"
                                                ]
                                            }

                                    ]
                                        }
                                    ],
                                    "rules": [
                                        {
                                            "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }

                ],
                                    "type": "tree"
                                },
                                {
                                    "conditions": [],
                                    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                    "type": "error"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
```

```
{
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                    {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"

  }
                ],
```

```
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

         }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
```

```
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",

      "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
```

```
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {

"endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
```

```
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
```

```
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
              "expect": {

        "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,

         "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
```

```
                "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
          },
          "HealthStatus": {
            "target":
```

```
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
         }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
```

```
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code> resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",

  "value": "DELETED",
            "documentation": ""
          }
        ]
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
       "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
          }
```

```
  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
```

```
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
```

```
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
```

```
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints </p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
```

```
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
           "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
```

```
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "InstanceUserSummaries"
            }
         }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
         "type": "structure",
         "members": {
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
    {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }

          },
            "Settings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
               "traits": {
                  "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
```

```
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
```

```
        "type": "union",

      "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
```

```
                    "min": 1
                },
                "smithy.api#required": {}
            }

    },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
```

```
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
```

```
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
         "type": "structure",
         "members": {
            "ProductUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
          },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) of the product user. "
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": " The domain name of the Active Directory that contains the user for whom to stop the product subscription. "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": " Metadata that describes the start product subscription operation. ",
                    "smithy.api#required": {}
                }

    }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {

```
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {

    "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
```

```
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
          "smithy.api#http": {

    "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns ...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns, c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
```

```
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
```

```
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
```

```
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
```

```
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]
```

```
permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
     - name: Run Scan code on pr ref
       run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
```

```
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
```

```
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
```

```
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
```

```
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
```

```
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
```

```
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
```

```
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
```

```
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

```
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}
```

```
type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
```

```
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
```

```
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
                 "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
               }
            },
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
                "traits": {
                    "smithy.api#documentation": "<p>An operator for filtering results.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
```

```
            "uri": "/subscription/GetServiceSettings"

          },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {}
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
              }
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
```

subscriptions-usage-operation.html\">Usage\n operation values in the License Manager User Guide. "
}
},
"ProductCode": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
"traits": {
"smithy.api#documentation": " The product code for the instance. For more information, see Usage\n operation values in the License Manager User Guide . "
}
},
"LastUpdatedTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The time in which the last discovery updated the instance details. "
}
},
"SubscriptionName": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the license subscription that the instance uses. "
}
},
"OsVersion": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The operating system software version that runs on your instance. "
}
},
"SubscriptionProviderCreateTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses. "
}
},
"SubscriptionProviderUpdateTime": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider. "
}
},
"DualSubscription": {
"target": "smithy.api#String",

```
            "traits": {
              "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
          },
```

```
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
       "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
```

```
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
```

```
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": {
                          "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
```

```
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

                                "ref": "UseFIPS"
                                        },
```

```
                true
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }

  ],
                "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"

        },
                            "supportsFIPS"
                        ]
                      }
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [

                         {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                      }
                    ]
```

```
                }
            ],
            "rules": [
                {
                    "conditions": [],

        "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],

                "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        }
    ],
    "type": "tree"
},
{

    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
```

```
                "conditions": [
                    {

            "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true

                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled
 but this partition does not support FIPS",
                "type": "error"
            }
        ],
        "type": "tree"
```

```
                },
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true

            ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {

                                  "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                              ]
                            }
                          ]
                        }
                      ],
                      "rules": [
                        {
                          "conditions": [],
                          "rules": [

    {
                                "conditions": [],
                                "endpoint": {
                                  "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                  "properties": {},
                                  "headers": {}
                                },
                                "type": "endpoint"
```

```
                                        }
                                    ],
                                    "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
],
"type": "tree"
}
]
},
```

```
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

  }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
```

```
                }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
```

```
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
```

```
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
                },
                {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
                      }
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                },

          {
                    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                    "expect": {
```

```
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",

            "expect": {
                "error": "Invalid Configuration: Missing Region"
              }
            }
          ],
          "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
          "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n          <ul>\n            <li>\n              <p>\n                <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n              <p>\n                <code>AmiID</code>\n              </p>\n            </li>\n
          <li>\n              <p>\n                <code>DualSubscription</code>\n              </p>\n            </li>\n
 <li>\n              <p>\n                <code>InstanceID</code>\n              </p>\n            </li>\n            <li>\n
  <p>\n                <code>InstanceType</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
    <code>ProductCode</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
 <code>Region</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
```

<code>Status</code>\n \n \n \n \n <code>UsageOperation</code>\n \n \n \n For each filter, you can use one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
}
},
"MaxResults": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
"traits": {
"smithy.api#documentation": " The maximum items to return in a request. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
"smithy.api#length": {
"min": 1,
"max": 16384
}
}
}
},
"traits": {
"smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
"type": "structure",
"members": {
"Instances": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
"traits": {
"smithy.api#documentation": " An array that contains instance objects. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
objects. "
}

```
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

              "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
```

```
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
match a certain subscription's name.</p>\n          <p>The valid names for this filter are:</p>\n          <ul>\n
<li>\n              <p>\n                  <code>Subscription</code>\n              </p>\n          </li>\n          </ul>\n
<p>The valid Operators for this filter are:</p>\n          <ul>\n          <li>\n              <p>\n
<code>contains</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
<code>equals</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
        <code>Notequal</code>\n              </p>\n          </li>\n          </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
```

```
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",
```

```
    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
        "smithy.api#http": {
          "method": "GET",
          "uri": "/tags/{resourceArn}"
        },
        "smithy.api#readonly": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
```

```
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
             },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
```

```
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
              },
              {

        "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
               "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
               "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

             "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
               "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
```

```
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
                }
            },
             "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
                }
            },
```

```
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
                }
            },
            "LastSuccessfulDataRetrievalTime": {
                "target": "smithy.api#String",
                "traits": {

                    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
```

```
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
```

```
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
 {
        "type": "list",
        "member": {
```

```
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
```

```
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
          "type": "map",
          "key":
{
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
```

```
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
```

```
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
{}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required":
{}
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowUpdate": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
          }
        }

      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }
```

```
    },
            "Status": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
               }
            },
            "StatusMessage": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
               "traits": {
                  "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
               }
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
             }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.82 aws-sdk-go-v2-service-internal-checksum 1.9.2

1.82.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

```
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
```

```
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
 // This member is required.
 Issuer *types.Issuer

 // License name.
 //
 // This member is required.
```

```
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
```

```
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
```

```
= addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
```

```
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
```

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

```
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
```

```
//
//   - Status
//
//
  - Fingerprint
//
//   - IssuerName
//
//   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
```

```
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
```

```
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
```

```
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string
```

```
 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
```

details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration. "
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": " The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory. "
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": " Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false. "
            }
          }
        },
        "traits": {
          "smithy.api#documentation": " Details about an Active Directory identity provider. "
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The domain name for the Active Directory. "
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": " A list of domain IPv4 addresses that are used for the Active Directory. ",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": " A list of domain IPv6 addresses that are used for the Active

```
Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                     "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
```

```
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

                 "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
           "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }

  },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
```

```
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

          },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) that identifies the identity provider to deregister. "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": " Metadata that describes the results of an identity provider operation. ",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },

```
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based subscriptions.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },

        "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
          "type": "structure",
```

```
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
            "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
            },
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Value of the filter.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
```

```
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

               "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
```

```
            "Status": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
              }

            },
            "FailureMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
              }
            },
            "OwnerAccountId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
```

```
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                  "smithy.api#required": {}
               }
            },
            "Products": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                  "smithy.api#required": {}
               }
            },
            "LastStatusCheckDate": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The date of the last status check.</p>"
               }
            },
            "StatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
               }
            },
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                  "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
 about the identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
         "type": "list",
```

```
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
 subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {

  "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
         },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
```

```
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
        },
        {

            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
    ],
    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager User Subscriptions",
            "arnNamespace": "license-manager-user-subscriptions",
            "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
```

```
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id",
                "content-length",
                 "x-api-key",
                "authorization",
                "x-amz-date",
                "x-amz-security-token",
                "Access-Control-Allow-Headers",
                "Access-Control-Allow-Methods",
                "Access-Control-Allow-Origin"
              ],
              "additionalExposedHeaders": [
                "x-amzn-errortype",
                "x-amzn-requestid",
                "x-amzn-trace-id"
              ]
            },
            "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
            "smithy.api#title": "AWS License Manager User Subscriptions",
            "smithy.rules#endpointRuleSet": {
              "version": "1.0",
              "parameters": {

      "Region": {
                  "builtIn": "AWS::Region",
                  "required": false,
                  "documentation": "The AWS region used to dispatch the request.",
                  "type": "string"
                },
                "UseDualStack": {
                  "builtIn": "AWS::UseDualStack",
                  "required": true,
                  "default": false,
                  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                  "type": "boolean"
                },
                "UseFIPS": {
                  "builtIn": "AWS::UseFIPS",
                  "required": true,
                  "default": false,
                  "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
```

```
            "type": "boolean"
          },
          "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
          }
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
```

```
                ]
              }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
          },
          {
            "conditions": [],
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },

"properties": {},
                "headers": {}
            },
            "type": "endpoint"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                 "argv": [
                   {
                     "ref": "Region"
                   }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },

true
                  ]
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

"fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        }
                      ]
                    },

                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
```

```
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }

        ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [

    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
```

```
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [

            {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
                  "type": "error"

}
              ],
              "type": "tree"
            },
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
```

```
                    },
                    true
                ]
            }
        ],
        "rules": [
            {

    "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]

     }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }

                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "DualStack is enabled but this partition does not support DualStack",
                "type": "error"
            }
        ],
        "type": "tree"
```

```
                        },
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                                "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ]
    },
    "smithy.rules#endpointTests": {
        "testCases": [
            {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                },
```

```
      "params": {
        "Region": "us-east-1",

    "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

  "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
 true,
        "UseDualStack": true
      }
```

```
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
"endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
```

```
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
```

```
                {
                  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint":
{
                      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                  },
                  "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

        "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                    }
```

```
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
```

```
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
          "type": "structure",
          "members": {
            "ProvisioningStatus": {
              "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
              }
            },
            "HealthStatus": {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
              "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
              }
            },
            "Ipv4Address": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
              }
```

```
            },
            "Ipv6Address": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
               }
            },
            "ServerType": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
               "traits": {
                  "smithy.api#documentation": "<p>The type of license server.</p>"
               }
            },
            "ServerEndpoint": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
               "traits": {
                  "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
               }
            },
            "StatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
               }
            },
            "LicenseServerEndpointId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
```

```
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
          },
          "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
          },
          "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
          },
          "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
              "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",

"value": "DELETED",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
```

```
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
              }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
```

```
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
```

```
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
     },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
         }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
```

```
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
      "type": "structure",

     "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
```

```
  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
```

```
{
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
```

```
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "ProductUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

      },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
               "method": "GET",
               "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
         "type": "structure",
         "members": {
            "ResourceArn": {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                  "smithy.api#httpLabel": {},
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
         "type": "structure",
         "members": {
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}

 }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
```

```
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
        }
    }
},
 "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
```

```
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }

 },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output":
 {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of a product for this user.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
        },
        "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
        },
        "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
```

```
        "type": "structure",
        "members": {
          "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {

          "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
                }

         },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
              }
            }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
```

```
                "smithy.api#length": {
                    "min": 1
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

 "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
```

```
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
          "type": "operation",
          "input":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

          {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
```

Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges in the Amazon Web Services Billing User Guide. </note>",
"smithy.api#http": {
"code": 200,
"method": "POST",

"uri": "/user/StartProductSubscription"
}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
"type": "structure",
"members": {
"Username": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The user name from the identity provider of the user. ",
"smithy.api#required": {}
}
},
"IdentityProvider": {
"target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
"traits": {
"smithy.api#documentation": " An object that specifies details for the identity provider. ",
"smithy.api#required": {}
}
},
"Product": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the user-based subscription product. Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> ",
"smithy.api#required": {}
}
},
"Domain": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The domain name of the Active Directory that contains the user for
whom to start the product subscription. "
}
},
"Tags": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
"traits": {
"smithy.api#documentation": " The tags that apply to the product subscription. "

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
            },
            "IdentityProvider":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
                }
            },
            "Domain": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
```

```
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {

      "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
  "<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
```

```
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
 {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {

           "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

          }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }

            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
```

```
            "traits": {
                "smithy.api#required": {}
            }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
          }
        }
      },

     "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
      }
   }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
```

```
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
```

```
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
```

```
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
```

```
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack,
options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
```

```
      python-version: [3.9]

  steps:
    - name: Checkout target
      uses: actions/checkout@v2
      with:
        path: sdkbase
        ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
```

```
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData
```

```
 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
```

```
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
```

```
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string
```

```
// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string
```

```
// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }
```

```
 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string
```

```
 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
```

```
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
```

```
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
```

```
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
```

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
```

```
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
```

CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
```

```
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
```

```
                "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

           "traits": {
                "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
              }
            },
            "SecretArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
               }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
              }
```

```
            },
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
         "type": "operation",
         "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/GetServiceSettings"

          },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
         "type": "structure",
         "members": {}
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
          }
        },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
        },
        "StatusMessage": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
          "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
          }
        },
        "HomeRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
           "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
```

```
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
        },
        "UsageOperation": {
          "target": "smithy.api#String",

   "traits": {
            "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
          }
        },
        "ProductCode": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
          "traits": {
            "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
```

```
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
        "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
        },
        "SubscriptionProviderUpdateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
        },
        "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
        },
        "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {
```

```
    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager Linux Subscriptions",
          "arnNamespace": "license-manager-linux-subscriptions",
          "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
```

```
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
```

```
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }

        ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]

}
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"

            }
          ],
```

```
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

                                    "ref": "UseFIPS"
                                        },
                                        true
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
```

```
                },
                true
            ]
        }

],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"

                },
                                                "supportsFIPS"
                                            ]
                                        }
                                    ]
                                },
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [

                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]
                                        }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],

    "rules": [
                                        {
```

```
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],

                        "type": "tree"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {

            "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {

                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "fn": "getAttr",
                                            "argv": [
```

```
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true

                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled
but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
```

```
                    },
                    true

                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {

                                    "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "rules": [

    {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"

                    }
                ],
                "type": "tree"
            },
```

```
                                    {
                                        "conditions": [],
                                        "error": "DualStack is enabled but this partition does not support DualStack",
                                        "type": "error"
                                    }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ],
        "type": "tree"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
```

```
    }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {

    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
          }
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
          }

  },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
```

```
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {

  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
       },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
```

```
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
```

```
                {

  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                    }

                },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                    }
                  },
                  "params": {
                    "Region": "us-iso-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
```

```
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },

    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
```

```
            }
          },
          {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
              "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {

      "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
              "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true,
              "Endpoint": "https://example.com"
            }
          },
          {
            "documentation": "Missing region",

          "expect": {
              "error": "Invalid Configuration: Missing Region"
            }
          }
        ],
        "version": "1.0"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
```

```
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled LinuxSubscriptionsDiscovery"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
            "SourceRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "OrganizationIntegration": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
                "traits": {
                    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n            </li>\n
 <li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n            </li>\n            <li>\n
  <p>\n                    <code>InstanceType</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
    <code>ProductCode</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
 <code>Region</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
 <code>Status</code>\n                </p>\n            </li>\n            <li>\n                <p>\n
 <code>UsageOperation</code>\n                </p>\n            </li>\n        </ul>\n        <p>For each filter, you can use
 one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n                <p>\n                    <code>contains</code>\n                </p>\n            </li>\n            <li>\n
 <p>\n                    <code>equals</code>\n                </p>\n            </li>\n            <li>\n
            <p>\n                    <code>Notequal</code>\n                </p>\n            </li>\n        </ul>"
                }
            },
```

```
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
          "type": "structure",
          "members": {
            "Instances": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
          },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n
```

```
        <code>Notequal</code>\n              </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
```

```
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

                 }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "tags": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
             },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
```

```
      "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSource": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
               "smithy.api#required": {}
             }
           },
           "SecretArn": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've
stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat
Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
               "smithy.api#required": {}
             }
           },
           "Tags": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
             "traits": {

            "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
             }
           }
         },
         "traits": {
           "smithy.api#input": {}
         }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
         "type": "structure",
         "members": {
           "SubscriptionProviderSource": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
             }
           },
           "SubscriptionProviderArn": {
             "target": "smithy.api#String",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
             }
           },
           "SubscriptionProviderStatus": {
             "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
             "traits": {
               "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
```

```
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
                }
            },
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
                }
            },
```

```
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {

                "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {
                  "name": "InProgress",
                  "value": "InProgress",
```

```
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }
        }
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
```

```
                {
                    "name": "RedHat",
                    "value": "RedHat",
                    "documentation": "RedHat subscription provider namespace"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
```

```
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
          "type": "map",
          "key":
{
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
          "type": "operation",
```

```
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

    "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input":
{}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
```

```
                "smithy.api#required":
{}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
            }
        },
        "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
        }

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }

  },
        "Status": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
          }
```

```
            },
            "StatusMessage": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
               "traits": {
                  "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
               }
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
             }
          }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
             "message": {
                "target": "smithy.api#String"
             }
          },
          "traits": {
             "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
             "smithy.api#error": "client"
          }
       }
    }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.83 ssl-client 1.37.0-r19

1.83.1 Available under license :

No license file was found, but licenses were detected in source scan.

+ * Licensed under GPLv2 or later, see file LICENSE in this source tree.

Found in path(s):
* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/0024-umount-Implement-O-option-to-unmount-by-mount-option.patch
No license file was found, but licenses were detected in source scan.

/*
* depmod - generate modules.dep
* Copyright (c) 2008 Bernhard Reutner-Fischer
- * Copyrihgt (c) 2008 Timo Teras <timo.teras@iki.fi>
+ * Copyrihgt (c) 2008-2015 Timo Teras <timo.teras@iki.fi>
* Copyright (c) 2008 Vladimir Dronnikov
*
* Licensed under GPLv2 or later, see file LICENSE in this source tree.
@@ -25,6 +25,24 @@
#include "modutils.h"
#include <sys/utsname.h> /* uname() */

+#define INDEX_MINCHAR 32
+#define INDEX_MAXCHAR 128
+
+typedef struct index_node {
+ char *prefix;
+ llist_t *values;
+ struct index_node *children[INDEX_MAXCHAR-INDEX_MINCHAR];
+} index_node;
+
+struct globals {
+ module_db db;
+ index_node *root_node;
+} FIX_ALIASING;
+#define G (*ptr_to_globals)

```
+#define INIT_G() do { \
+ SET_PTR_TO_GLOBALS(xzalloc(sizeof(G))); \
+} while (0)
+
/*
* Theory of operation:
* - iterate over all modules and record their full path
@@ -53,18 +71,12 @@ static int FAST_FUNC parse_module(struct recursive_state
 *state,

for (ptr = image; ptr < image + len - 10; ptr++) {
if (is_prefixed_with(ptr, "depends=")) {
-   char *u;
-
ptr += 8;
-   for (u = ptr; *u; u++)
-    if (*u == '-')
-     *u = '_';
-   ptr += string_to_llist(ptr, &e->deps, ",");
+   string_to_llist(replace_underscores(ptr), &e->deps, ",");
} else if (ENABLE_FEATURE_MODUTILS_ALIAS
&& is_prefixed_with(ptr, "alias=")
) {
-   llist_add_to(&e->aliases, xstrdup(ptr + 6));
-   ptr += strlen(ptr);
+   llist_add_to(&e->aliases, replace_underscores(xstrdup(ptr + 6)));
} else if (ENABLE_FEATURE_MODUTILS_SYMBOLS
&& is_prefixed_with(ptr, "__ksymtab_")
) {
@@ -74,9 +86,10 @@ static int FAST_FUNC parse_module(struct recursive_state *state,
) {
continue;
}
-   llist_add_to(&e->symbols, xstrdup(ptr));
-   ptr += strlen(ptr);
- }
+   llist_add_to(&e->symbols, xasprintf("symbol:%s", ptr));
+  } else
+   continue;
+  ptr += strlen(ptr);
}
free(image);

@@ -108,12 +121,6 @@ static void order_dep_list(module_db *modules, module_entry *start, llist_t
 *add
}
}
```

-static void xfreopen_write(const char *file, FILE *f)
-{
- if (freopen(file, "w", f) == NULL)
- bb_perror_msg_and_die("can't open '%s'", file);
-}
-
//usage:#if !ENABLE_MODPROBE_SMALL
//usage:#define depmod_trivial_usage "[-n] [-b BASE] [VERSION] [MODFILES]..."
//usage:#define depmod_full_usage "\n\n"
@@ -167,6 +174,169 @@ enum {
OPT_C = (1 << 9), /* -C,--config etc_modules_conf: ignored */

Found in path(s):
* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/0018-depmod-support-generating-kmod-binary-index-files.patch
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Natanael Copa <natanael.copa@gmail.com>
* All rights reserved.
*
* This program is free software; you can redistribute it and/or modify it
* under the terms of the GNU General Public License version 2 as published
* by the Free Software Foundation. See http://www.gnu.org/ for details.
*
*/

Found in path(s):
* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/bbsuid.c

1.84 libapk2 2.14.9-r3

1.84.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license
document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share

and change it. By contrast, the GNU General Public License is intended to
guarantee your freedom to share and change free software--to make sure the
software is free for all its users. This General Public License applies to
most of the Free Software Foundation's software and to any other program whose
authors commit to using it. (Some other Free Software Foundation software
is covered by the GNU Lesser General Public License instead.) You can apply
it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our
General Public Licenses are designed to make sure that you have the
freedom
to distribute copies of free software (and charge for this service if you
wish), that you receive source code or can get it if you want it, that you
can change the software or use pieces of it in new free programs; and that
you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to
deny you these rights or to ask you to surrender the rights. These restrictions
translate to certain responsibilities for you if you distribute copies of
the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or
for a fee, you must give the recipients all the rights that you have. You
must make sure that they, too, receive or can get the source code. And you
must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2)
offer you this license which gives you legal permission to copy, distribute
and/or modify the software.

Also, for
each author's protection and ours, we want to make certain that
everyone understands that there is no warranty for this free software. If
the software is modified by someone else and passed on, we want its recipients
to know that what they have is not the original, so that any problems introduced
by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We
wish to avoid the danger that redistributors of a free program will individually
obtain patent licenses, in effect making the program proprietary. To prevent
this, we have made it clear that any patent must be licensed for everyone's
free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification
follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice
placed by the copyright holder saying it may be distributed under
 the terms
of this General Public License. The "Program", below, refers to any such program
or work, and a "work based on the Program" means either the Program or any
derivative work under copyright law: that is to say, a work containing the
Program or a portion of it, either verbatim or with modifications and/or translated
into another language. (Hereinafter, translation is included without limitation
in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered
by this License; they are outside its scope. The act of running the Program
is not restricted, and the output from the Program is covered only if its
contents constitute a work based on the Program (independent of having been
made by running the Program). Whether that is true depends on what the Program
does.

1. You may copy and distribute verbatim copies of the Program's source code
as you receive it, in any medium, provided that you conspicuously
 and appropriately
publish on each copy an appropriate copyright notice and disclaimer of warranty;
keep intact all the notices that refer to this License and to the absence
of any warranty; and give any other recipients of the Program a copy of this
License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you
may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it,
thus forming a work based on the Program, and copy and distribute such modifications
or work under the terms of Section 1 above, provided that you also meet all
of these conditions:

a) You must cause the modified files to carry prominent notices stating that
you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or
in part contains or is derived from the Program or any part thereof, to be
licensed as a whole at no charge
 to all third parties under the terms of this
License.

c) If the modified program normally reads commands interactively when run,
you must cause it, when started running for such interactive use in the most
ordinary way, to print or display an announcement including an appropriate
copyright notice and a notice that there is no warranty (or else, saying that
you provide a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this License.

(Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.
But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it
with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means

all
the source code for all modules it contains, plus any associated interface
definition files, plus the scripts used to control compilation and installation
of the executable. However, as a special exception, the source code distributed
need not include anything that is normally distributed (in either source or
binary form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component itself
accompanies the executable.

If distribution of executable or object code is made by offering access to
copy from a designated place, then offering equivalent access to copy the
source code from the same place counts as distribution of the source code,
even though third parties are not compelled to copy the source along with
the object code.

4. You may not copy, modify, sublicense, or distribute the Program except
as expressly provided under this License. Any attempt otherwise to copy, modify,
sublicense or distribute the Program
is void, and will automatically terminate
your rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses terminated
so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed
it. However, nothing else grants you permission to modify or distribute the
Program or its derivative works. These actions are prohibited by law if you
do not accept this License. Therefore, by modifying or distributing the Program
(or any work based on the Program), you indicate your acceptance of this License
to do so, and all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program),
the recipient automatically receives a license from the original licensor
to copy, distribute or modify the Program subject to these terms and conditions.
You may
not impose any further restrictions on the recipients' exercise of
the rights granted herein. You are not responsible for enforcing compliance
by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement
or for any other reason (not limited to patent issues), conditions are imposed
on you (whether by court order, agreement or otherwise) that contradict the
conditions of this License, they do not excuse you from the conditions of
this License. If you cannot distribute so as to satisfy simultaneously your
obligations under this License and any other pertinent obligations, then as
a consequence you may not distribute the Program at all. For example, if a
patent license would not permit royalty-free redistribution of the Program
by all those who receive copies directly or indirectly through you, then the

only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse

of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best
way to achieve this is to make it free software
which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS

FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a
copy of the GNU General Public License along with
this program; if not, write to the Free Software Foundation, Inc., 51 Franklin
Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when
it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes
with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software,
and you are welcome to redistribute it under certain conditions; type `show
c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may be
called something other than `show w' and `show c'; they could even be mouse-clicks
or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school,
if any, to sign a "copyright
disclaimer" for the program, if necessary. Here
is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision'
(which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General
Public License does not permit incorporating your program into proprietary
programs. If your program is a subroutine library, you may consider it more
useful to permit linking proprietary applications with the library. If this
is what you want to do, use the GNU Lesser General Public License instead
of this License.

1.85 github.com/aws/aws-sdk-go-v2/internal/endpoints/v2 2.7.11

1.85.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
```

```
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag
```

```
 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
```

```
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
```

```
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
```

```
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength

```
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
```

```
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense
```

```
 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
```

```
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
 addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
```

```
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
```

```
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
```

```
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {
```

```
 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
```

```
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
```

ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
        "type": "structure",
        "members": {
            "DirectoryId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
                "traits": {
                    "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
                }
            },
            "ActiveDirectorySettings":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
                }
            },
            "ActiveDirectoryType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
                }
            },
            "IsSharedActiveDirectory": {
                "target": "smithy.api#Boolean",
                "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
```

```
            "DomainName": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
              }
            },
            "DomainIpv4List": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainIpv6List": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
              }
            },
            "DomainNetworkSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
             "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {

     "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider for the user.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
```

```
state of the resource.</p>",
            "smithy.api#error": "server"
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
{
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
```

```
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
              },
              {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
              }
            ],
            "traits": {
              "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
              "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
              "smithy.api#idempotent": {}
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
            "type": "structure",
            "members": {
              "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                  "smithy.api#required": {}
                }
              },
              "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                  "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                  "smithy.api#required": {}

    }
              }
            },
            "traits": {
              "smithy.api#input": {}
            }
          },
          "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
            "type": "structure",
            "members": {
              "LicenseServerEndpoint": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

       },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
          "smithy.api#http": {
```

```
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
```

```
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
```

```
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
                 }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
                }
            }
        },
        "traits": {
```

```
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
              }
            }
          },

        "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
          "type": "structure",
          "members": {
            "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members":
{
            "Attribute": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
              }
            },
```

```
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Value of the filter.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
            "ActiveDirectoryIdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
```

about an identity provider. ",
"smithy.api#required": {}
}
},
"Settings": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
"traits": {
"smithy.api#documentation": " The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints. ",

"smithy.api#required": {}
}
},
"Product": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the user-based subscription product. ",
"smithy.api#required": {}
}
},
"Status": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The status of the identity provider. ",
"smithy.api#required": {}
}
},
"IdentityProviderArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The Amazon Resource Name (ARN) of the identity provider. "
}

},
"FailureMessage": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The failure message associated with an identity provider. "
}
},
"OwnerAccountId": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The AWS Account ID of the owner of this resource. "
}
}
},

```
        "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
```

```
            },
            "OwnerAccountId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                  "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                  "smithy.api#required": {}
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
```

```
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
                }
            },
            "AssociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
                }
            },
            "DisassociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
```

```
"type": "service",
"version": "2018-05-10",
"operations": [
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
```

```
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
        },
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
```

```
    "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
```

```
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },

            "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
```

```
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                 "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },

true
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {

"fn": "booleanEquals",
                        "argv": [
```

```
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        }
                      ]
                    },

                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]
                        }

                  ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }

        ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS and DualStack are enabled, but this partition does not support one or
```

```
both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [

    {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [

                {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {

              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
```

```
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                "fn": "booleanEquals",
                "argv": [
                    {
                    "ref": "UseDualStack"
                    },
                    true
                ]
                }
            ],
            "rules": [
                {

    "conditions": [
                        {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                            "fn": "getAttr",
                            "argv": [
                                {
                                "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]

 }
                        ]
                        }
                    ],
                    "rules": [
```

```
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }

                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
"smithy.rules#endpointTests": {
    "testCases": [
```

```
                {
                  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",

              "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
                  "expect": {

  "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
```

```
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
```

```
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
```

```
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
```

```
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

 "UseDualStack": false,
          "Endpoint": "https://example.com"
        }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
```

```
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
```

```
            "ProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
                }
            },
            "HealthStatus": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The type of license server.</p>"
            }
        },
        "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
            }
        },
        "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
        },
        "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
        },
        "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
        },
        "CreationTime": {
            "target": "smithy.api#Timestamp",
```

```
    "traits": {
                "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
```

```
                "documentation": ""
              },
              {
                "name": "DELETED",

"value": "DELETED",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
```

```
            },
            "ServerSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                  "smithy.api#required": {}
               }

  }
         },
         "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
```

```
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
               "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
               }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources that contain details about the Active Directory identity providers that meet the request criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/instance/ListInstances"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
```

```
      "type": "structure",

    "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },

    {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListProductSubscriptions"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
              }
            }
          },
```

```
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                 "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
```

```
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }

 },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummaries": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output":
 {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
                  "smithy.api#required": {}
               }
            },
```

```
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

 "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
          }
        },
```

```
            "SubscriptionEndDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
```

```
            }

        },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
```

```
            "message": {
               "target": "smithy.api#String"

            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
         "type": "structure",
         "members": {
            "SecretId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                  "smithy.api#length": {
                     "min": 1
                  }
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
         "type": "string",
         "traits": {
            "smithy.api#length": {
               "min": 5,
               "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
         "type": "structure",
         "members": {
            "Endpoint": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
               }
```

```
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
          "type": "union",

         "members": {
            "RdsSalSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "RDS_SAL",
                "value": "RDS_SAL",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
 {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
          }
        },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }

    },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
      "type": "operation",
      "input":
{
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

         {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                  "smithy.api#required": {}
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
```

```
            "target": "smithy.api#String",
             "traits": {
               "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
               "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
               "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
```

```
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {

"min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
```

```
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {

           "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
          "smithy.api#http": {
```

```
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

          }
          },
          "TagKeys": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
                }
            },
            "UpdateSettings": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
```

```
            },
            "SecurityGroupId": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
               "traits": {
                  "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
               }
            }
         },

         "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
```

```
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
```

```
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region:
```

```
        region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
```

```
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string
```

```
// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
```

```
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {
```

```
// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
```

```
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
```

```
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options)) (*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}
```

```
type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {
```

```
 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
```

```
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
```

```
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
```

```
    {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/DeregisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
```

```
            }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
              }
            },
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

           "traits": {
                "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
              }
            },
            "SecretArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
```

```
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
               "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
        "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
        },
```

```
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
```

```
"type": "service",
"version": "2018-05-10",
"operations": [
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
     },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
         "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
```

```
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager Linux Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {
              "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "String"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                 "type": "Boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "Boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "String"
              }
```

```
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }

      ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]

}
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
```

```
            },
            {
              "conditions": [],
              "endpoint": {
                "url": {
                  "ref": "Endpoint"
                },
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ],
          "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {

    "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
```

```
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {

                    "ref": "UseFIPS"
                        },
                        true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }

],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"

            },
                                "supportsFIPS"
                              ]
                            }
                          ]
                        },
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
```

```
                                        "argv": [

                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],

    "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],

                    "type": "tree"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {

        "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
```

```
                {
                  "ref": "UseFIPS"
                },
                true
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {

        "fn": "booleanEquals",
                  "argv": [
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                      ]
                    },
                    true

                ]
                }
              ],
              "rules": [
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                }
              ],
              "type": "tree"
```

```
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled
but this partition does not support FIPS",
                  "type": "error"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true

          ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {

                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]
                        }
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "rules": [
```

```
        {
                                                "conditions": [],
                                                "endpoint": {
                                                   "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                   "properties": {},
                                                   "headers": {}
                                                },
                                                "type": "endpoint"
                                             }
                                          ],
                                          "type": "tree"

                                    }
                                       ],
                                       "type": "tree"
                                    },
                                    {
                                       "conditions": [],
                                       "error": "DualStack is enabled but this partition does not support DualStack",
                                       "type": "error"
                                    }
                                 ],
                                 "type": "tree"
                              },
                              {
                                 "conditions": [],
                                 "rules": [
                                    {
                                       "conditions": [],
                                       "endpoint": {
                                          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                       },
                                       "type": "endpoint"
                                    }
                                 ],
                                 "type": "tree"
                              }
                           ],
                           "type": "tree"
                        }
                     ],
                     "type": "tree"
                  },
                  {
```

```
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
              }
            ],
            "type": "tree"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

        }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {

         "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
```

```
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

      },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

      },
      "params": {
        "Region": "us-gov-east-1",
```

```
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
```

```
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },

    {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "Missing region",
```

```
                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                }
                }
            ],
            "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled LinuxSubscriptionsDiscovery"
          },
          {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
      "type": "structure",
      "members": {
        "SourceRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "OrganizationIntegration": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
          "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
```

```
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
          "type": "structure",
          "members": {
            "Instances": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
              "traits": {
```

```
                "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
 objects.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                  }
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
            "type": "structure",
            "members": {
              "Subscriptions": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
```

```
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "RegisteredSubscriptionProviders"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSources": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

  "traits": {
                    "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
```

```
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
           }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "EQUAL",
              "value": "Equal",
              "documentation": "Equal operator"
            },
            {
              "name": "NOT_EQUAL",
              "value": "NotEqual",
              "documentation": "Not equal operator"
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
```

```
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {

                    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
```

```
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
              }
            },
            "SubscriptionProviderArn": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
              }
            },
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
              }
            },
            "SecretArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
```

```
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
```

```
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
```

```
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with
Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace.
Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing
from Amazon EC2 or BYOS.</p>"
                }
            },
            "InstanceCount": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
                "traits": {
                    "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
```

```
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
```

```
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

  "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
{}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
```

```
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                    "smithy.api#required":
{}
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
                    "smithy.api#required": {}
                }
            },
            "AllowUpdate": {
                "target": "smithy.api#Boolean",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
                }
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
```

```
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
            }

    },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
            }
         }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
      }
    }
  }
```

}

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.86 clientmodel 0.6.2

1.86.1 Available under license :

Data model artifacts for Prometheus.
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.87 skalibs-libs 2.14.4.0-r0

1.87.1 Available under license :

Copyright (c) 2011-2025 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.88 aws-sdk-go-v2-service-ssooidc 1.35.3

1.88.1 Available under license :

GetLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware

```
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
```

```
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
```

```
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
```

```
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
```

```
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning

Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
  - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
```

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32
```

```
 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
```

```
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
 return err
 }
 if err = addRetry(stack, options); err != nil {
 return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
 err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
```

```
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
```

```
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
```

```
        "DomainIpv6List": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
            "smithy.api#length": {
              "min": 1,
              "max": 2
            }
          }
        },
        "DomainCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
          }
        },
        "DomainNetworkSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
            "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
        "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
```

```
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
             "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
               "smithy.api#required": {}
             }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
          "smithy.api#error": "server"
        }

    },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

          {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
         "type": "structure",
         "members":
 {
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
              }
            },
            "LicenseServerEndpointArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code> resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
          "type": "union",
          "members": {
            "SecretsManagerCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials needed for user administration in
```

```
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"
```

```
        },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
              "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
```

```
    },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/DeregisterIdentityProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
          }
```

```
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
               }
            },
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },

   "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
         "type": "structure",
         "members": {
            "IdentityProviderSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
         "type": "string",
         "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
         "type": "operation",
         "input": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
```

```
identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
                 }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
```

```
    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
```

```
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
        "type": "union",
        "members": {
          "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

               "smithy.api#required": {}
            }
          },
          "Product": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
            }

          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

                  "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
```

```
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
     },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
```

```
            "name": "license-manager-user-subscriptions"
          },
          "aws.protocols#restJson1": {},
          "smithy.api#cors": {
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id",
              "content-length",
               "x-api-key",
              "authorization",
              "x-amz-date",
              "x-amz-security-token",
              "Access-Control-Allow-Headers",
              "Access-Control-Allow-Methods",
              "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager User Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {

      "Region": {
                  "builtIn": "AWS::Region",
                  "required": false,
                  "documentation": "The AWS region used to dispatch the request.",
                  "type": "string"
                },
                "UseDualStack": {
                  "builtIn": "AWS::UseDualStack",
                  "required": true,
                  "default": false,
                  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                  "type": "boolean"
                },
                "UseFIPS": {
```

```
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
              },
              "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
              }
            },
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Endpoint"
                      }
                    ]

}
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseFIPS"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [
```

```
{
                        "fn": "booleanEquals",
                        "argv": [
                          {
                            "ref": "UseDualStack"
                          },
                          true
                        ]
                      }
                    ],
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                    "type": "error"
                  },
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": {
                        "ref": "Endpoint"
                      },

                "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [
                  {
                    "fn": "isSet",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "aws.partition",
                         "argv": [
                          {
                            "ref": "Region"
                          }
```

```
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

true

```
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
```

"fn": "booleanEquals",

```
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          }
                        ]
                      },
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                      ]
                    }

                ]
              }
            ],
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }

        ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
            "type": "error"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
```

```
{
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                    {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsFIPS"
                                    ]
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"

  }
                ],
```

```
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

         }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
```

```
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
```

```
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",

      "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
```

```
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {

"endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS":
```

```
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS":
false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
```

```
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
              "expect": {

        "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
              },
              "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,

         "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
              "expect": {
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
```

```
                "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                "expect": {
                  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                 }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
          },
          "HealthStatus": {
            "target":
```

```
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
                }
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                }
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
```

```
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {

          "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",

"value": "DELETED",
            "documentation": ""
          }
        ]
      }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
       "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
          }
```

```
  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
           {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
```

```
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
```

```
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
```

```
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints </p>",
          "smithy.api#http": {
```

```
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
```

```
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
```

```
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
```

```
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
           "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
        },
```

```
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

  "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }

    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
{}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
            }
          },
          "SubscriptionStartDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
          },
          "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
         "type": "structure",
         "members": {
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                  "smithy.api#required": {}
               }
            },
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                  "smithy.api#required": {}
               }

          },
            "Settings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
               "traits": {
                  "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
```

```
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
```

```
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
```

```
        "type": "union",

      "members": {
          "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
              "smithy.api#length": {
```

```
                    "min": 1
                },
                "smithy.api#required": {}
            }

    },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
```

```
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
             "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
              "smithy.api#required": {}
            }
          },
```

```
            "Domain": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
               }
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
         "type": "structure",
         "members": {
            "ProductUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
               "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
          },
         "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> "
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": " The Amazon Resource Name (ARN) of the product user. "
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": " The domain name of the Active Directory that contains the user for whom to stop the product subscription. "
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": " Metadata that describes the start product subscription operation. ",
                    "smithy.api#required": {}
                }

    }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {

```
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {

        "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
```

```
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
          "smithy.api#http": {

   "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }
```

```
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
          }
        },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
```

```
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
```

```
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
```

```
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
```

```
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack,
```

```
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]
```

```
permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
     - name: Run Scan code on pr ref
       run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
```

```
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
```

```
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
```

```
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
```

```
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
```

```
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
}
```

```
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
```

```
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription
```

```
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
```

```
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
```

```
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
```

```
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
```

```
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

```
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}
```

```
type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
```

```
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
```

```
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
                 "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
          "type": "structure",
          "members": {
            "Name": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
               }
            },
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
```

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
          "type": "operation",
          "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
```

```
            "uri": "/subscription/GetServiceSettings"

          },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
          "type": "structure",
          "members": {}
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
              }
            },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
              }
            }
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
          "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
          },
          "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
              "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
          },
          "AccountID": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
          },
          "Region": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
          },
          "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
              "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
```

```
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
                }
            },
            "ProductCode": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
                "traits": {
                    "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
                }
            },
            "LastUpdatedTime": {
                "target": "smithy.api#String",
                 "traits": {
                    "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
                }
            },
            "SubscriptionName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
                }
            },
            "OsVersion": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
                }
            },
            "SubscriptionProviderCreateTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
                }
            },
            "SubscriptionProviderUpdateTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
                }
            },
            "DualSubscription": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
            }
        },
        "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager.</p>"
            }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
```

```
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ],
    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
            "additionalAllowedHeaders": [
                "*",
                "content-type",
                "x-amz-content-sha256",
                "x-amz-user-agent",
                "x-amzn-platform-id",
                "x-amzn-trace-id"
            ],
            "additionalExposedHeaders": [
```

```
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "String"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
               "type": "Boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "Boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "String"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
```

```
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": {
                          "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
```

```
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {

                                "ref": "UseFIPS"
                                        },
```

```
                true
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }

  ],
                "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"

        },
                            "supportsFIPS"
                        ]
                      }
                    ]
                  },
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [

                         {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                      }
                    ]
```

```
                }
            ],
            "rules": [
                {
                    "conditions": [],

        "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],

                "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        }
    ],
    "type": "tree"
},
{

    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
```

```
                "conditions": [
                  {

            "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true

                ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS is enabled
 but this partition does not support FIPS",
                "type": "error"
              }
            ],
            "type": "tree"
```

```
                    },
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true

                ]
                        }
                      ],
                      "rules": [
                        {
                          "conditions": [
                            {
                              "fn": "booleanEquals",
                              "argv": [
                                true,
                                {
                                  "fn": "getAttr",
                                  "argv": [
                                    {

                                      "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                  ]
                                }
                              ]
                            }
                          ],
                          "rules": [
                            {
                              "conditions": [],
                              "rules": [

    {
                                  "conditions": [],
                                  "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                  },
                                  "type": "endpoint"
```

```
                                        }
                                    ],
                                    "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
],
"type": "tree"
}
]
},
```

```
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

  }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
```

```
                }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
              }
             },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": false,
              "UseDualStack": true
```

```
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

     },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
```

```
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
                },
                {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                },
                {
                    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
                      }
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                },

          {
                    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                    "expect": {
```

```
                "endpoint": {
                  "url": "https://example.com"
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",

            "expect": {
                "error": "Invalid Configuration: Missing Region"
              }
            }
          ],
          "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
```

```
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
              "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
        "type": "structure",
        "members": {
          "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
              "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Instances"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n          <ul>\n            <li>\n              <p>\n                <code>AccountID</code>\n
 </p>\n            </li>\n            <li>\n              <p>\n                <code>AmiID</code>\n              </p>\n            </li>\n
          <li>\n              <p>\n                <code>DualSubscription</code>\n              </p>\n            </li>\n
 <li>\n              <p>\n                <code>InstanceID</code>\n              </p>\n            </li>\n            <li>\n
  <p>\n                <code>InstanceType</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
    <code>ProductCode</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
 <code>Region</code>\n              </p>\n            </li>\n            <li>\n              <p>\n
```

<code>Status</code>\n \n \n \n \n <code>UsageOperation</code>\n \n \n \n For each filter, you can use one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
}
},
"MaxResults": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
"traits": {
"smithy.api#documentation": " The maximum items to return in a request. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
"smithy.api#length": {
"min": 1,
"max": 16384
}
}
}
},
"traits": {
"smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
"type": "structure",
"members": {
"Instances": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
"traits": {
"smithy.api#documentation": " An array that contains instance objects. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
objects. "
}

```
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Subscriptions"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

              "traits": {
                    "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
```

```
that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n
 <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
match a certain subscription's name.</p>\n          <p>The valid names for this filter are:</p>\n          <ul>\n
<li>\n              <p>\n                  <code>Subscription</code>\n              </p>\n          </li>\n          </ul>\n
<p>The valid Operators for this filter are:</p>\n          <ul>\n          <li>\n              <p>\n
<code>contains</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
<code>equals</code>\n              </p>\n          </li>\n          <li>\n              <p>\n
        <code>Notequal</code>\n              </p>\n          </li>\n          </ul>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 16384
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
```

```
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",
```

```
    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
        "smithy.api#http": {
          "method": "GET",
          "uri": "/tags/{resourceArn}"
        },
        "smithy.api#readonly": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
```

```
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
             },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
```

```
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
              },
              {

        "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
             "traits": {
               "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
               "smithy.api#required": {}
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
               "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
               "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
```

```
 the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
                }
            },
             "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
                }
            },
```

```
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
                }
            },
            "LastSuccessfulDataRetrievalTime": {
                "target": "smithy.api#String",
                "traits": {

                    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
```

```
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
```

```
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
 {
        "type": "list",
        "member": {
```

```
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
```

```
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
          "type": "map",
          "key":
{
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
```

```
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

  "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tagKeys": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
```

```
\n\t\t\tresource.</p>",
                    "smithy.api#httpQuery": "tagKeys",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input":
 {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required":
{}
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
          }
        },
        "AllowUpdate": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
          }
        }

      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
          }
```

```
    },
            "Status": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
               }
            },
            "StatusMessage": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
               "traits": {
                  "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
               }
            },
            "HomeRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
               }
             }
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
         }
      }
   }
}
```


Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.89 github.com/aws/smithy-go 1.23.1

1.89.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

1.90 musl-utils 1.2.5-r10

1.90.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

Copyright 2005-2020 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
 LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Authors/contributors include:

A. Wilcox
Ada Worcester
Alex Dowad
Alex Suykov
Alexander Monakov
Andre McCurdy
Andrew Kelley
Anthony G. Basile
Aric Belsito
Arvid Picciani
Bartosz Brachaczek
Benjamin Peterson
Bobby Bingham
Boris Brezillon
Brent Cook
Chris Spiegel
Clment Vasseur
Daniel Micay
Daniel Sabogal
Daurnimator
David Carlier
David Edelsohn
Denys Vlasenko

Dmitry Ivanov
Dmitry V. Levin
Drew DeVault
Emil Renner Berthing
Fangrui Song
Felix Fietkau
Felix Janda
Gianluca Anzolin
Hauke Mehrtens
He X
Hiltjo Posthuma
Isaac Dunham
Jaydeep Patil
Jens Gustedt
Jeremy Huntwork
Jo-Philipp Wich
Joakim Sindholt
John Spencer
Julien Ramseier
Justin Cormack
Kaarle Ritvanen
Khem Raj
Kylie McClain
Leah Neukirchen
Luca Barbato
Luka
Perkov
M Farkas-Dyck (Strake)
Mahesh Bodapati
Markus Wichmann
Masanori Ogino
Michael Clark
Michael Forney
Mikhail Kremnyov
Natanael Copa
Nicholas J. Kain
orc
Pascal Cuoq
Patrick Oppenlander
Petr Hosek
Petr Skocik
Pierre Carrier
Reini Urban
Rich Felker
Richard Pennington
Ryan Fairfax
Samuel Holland
Segev Finer

Shiz
sin
Solar Designer
Stefan Kristiansson
Stefan O'Rear
Szabolcs Nagy
Timo Ters
Trutz Behn
Valentin Ochs
Will Dietz
William Haddon
William Pitcock

Portions of this software are derived from third-party works licensed
under terms compatible with the above MIT license:

The TRE regular expression implementation (src/regex/reg* and
src/regex/tre*) is Copyright 2001-2008 Ville Laurikari and licensed
under a 2-clause BSD license (license text in the source files). The
included version has been heavily modified by Rich Felker in 2012, in
the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (src/math/* and src/complex/*) is
Copyright 1993,2004
 Sun Microsystems or
Copyright 2003-2011 David Schultz or
Copyright 2003-2009 Steven G. Kargl or
Copyright 2003-2009 Bruce D. Evans or
Copyright 2008 Stephen L. Moshier or
Copyright 2017-2018 Arm Limited
and labelled as such in comments in the individual source files. All
have been licensed under extremely permissive terms.

The ARM memcpy code (src/string/arm/memcpy.S) is Copyright 2008
The Android Open Source Project and is licensed under a two-clause BSD
license. It was taken from Bionic libc, used on Android.

The AArch64 memcpy and memset code (src/string/aarch64/*) are
Copyright 1999-2019, Arm Limited.

The implementation of DES for crypt (src/crypt/crypt_des.c) is
Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (src/crypt/crypt_blowfish.c) was
originally written by Solar Designer and placed into the public
domain. The code also comes with a fallback permissive license for use
in jurisdictions that may not recognize the

public domain.

The smoothsort implementation (src/stdlib/qsort.c) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the ellcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contibutors listed
above. Details on authorship of individual files
can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/* and arch/*/bits/*) and crt files intended to be linked into applications (crt/*, ldso/dlstart.c, and arch/*/crt_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham
John Spencer
Nicholas J. Kain
Rich Felker
Richard Pennington
Stefan Kristiansson
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not

to be subject to copyright, resulting in confusion over whether it negated
the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

1.91 x-sys 0.37.0

1.91.1 Available under license :

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.92 github.com/aws/aws-sdk-go-v2/internal/ini 1.8.4

1.92.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
```

```
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
```

```
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
```

```
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
```

```
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent

AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

```
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {
```

```
 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err =
addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
```

```
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
```

```
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
```

```
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {
```

```
 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
```

```
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
 return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
 return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
 return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
```

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
```

```
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
        "type": "structure",
        "members": {
          "DirectoryId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
            "traits": {
              "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
          },
          "ActiveDirectorySettings":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

           "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
```

```
        "members": {
            "DomainName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
                }
            },
            "DomainIpv4List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainIpv6List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
                }
            },
            "DomainNetworkSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
             "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
      },
      "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",
```

```
            "smithy.api#required": {}
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
          }

 },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

         {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":
{
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
          }
        },
        "LicenseServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                    "smithy.api#required": {}

  }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
```

```
        "LicenseServerEndpoint": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
          "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

     },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

      },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
```

```
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {

    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
```

```
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
              }
            },
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
               }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
              }
            }
          },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },

    "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
```

```
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
```

```
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
          }

        },
        "FailureMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
          }
        },
        "OwnerAccountId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
          }
        }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
 products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
```

```
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Status": {
              "target": "smithy.api#String",
              "traits": {

                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
```

```
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
      },
```

```
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
         },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
         },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
        },
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
```

```
    "Region": {
                    "builtIn": "AWS::Region",
                    "required": false,
                    "documentation": "The AWS region used to dispatch the request.",
                    "type": "string"
                },
                "UseDualStack": {
                    "builtIn": "AWS::UseDualStack",
                    "required": true,
                    "default": false,
                    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                    "type": "boolean"
                },
                "UseFIPS": {
                    "builtIn": "AWS::UseFIPS",
                    "required": true,
                    "default": false,
                    "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                    "type": "boolean"
                },
                "Endpoint": {
                    "builtIn": "SDK::Endpoint",
                    "required": false,
                    "documentation": "Override the endpoint used to send this request",
                    "type": "string"
                }
            },
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Endpoint"
                                }
                            ]

    }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
```

```
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
                "endpoint": {
                  "url": {
                    "ref": "Endpoint"
                  },

            "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
```

```
                {
                  "ref": "Region"
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {
```

```
        "fn": "booleanEquals",
                                    "argv": [
                                      true,
                                      {
                                        "fn": "getAttr",
                                        "argv": [
                                          {
                                            "ref": "PartitionResult"
                                          },
                                          "supportsFIPS"
                                        ]
                                      }
                                    ]
                                  },

                                  {
                                    "fn": "booleanEquals",
                                    "argv": [
                                      true,
                                      {
                                        "fn": "getAttr",
                                        "argv": [
                                          {
                                            "ref": "PartitionResult"
                                          },
                                          "supportsDualStack"
                                        ]
                                      }

                                ]
                                    }
                                  ],
                                  "rules": [
                                    {
                                      "conditions": [],
                                      "endpoint": {
                                        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                      },
                                      "type": "endpoint"
                                    }

          ],
                                  "type": "tree"
                                },
                                {
```

```
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [

        {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [

                {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "endpoint": {

                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
```

```
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled but this partition does not support FIPS",
                    "type": "error"

}
                ],
                "type": "tree"
            },
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {

        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]

        }
                            ]
                        }
```

```
                                    ],
                                    "rules": [
                                        {
                                            "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }

                                    ],
                                    "type": "tree"
                                },
                                {
                                    "conditions": [],
                                    "error": "DualStack is enabled but this partition does not support DualStack",
                                    "type": "error"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ]
},
```

```
        "smithy.rules#endpointTests": {
          "testCases": [
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
              },
              "params": {
                "Region": "us-east-1",

          "UseFIPS": true,
                "UseDualStack": false
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
              "expect": {
                "endpoint": {
                  "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
                }
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true
              }
            },
            {
              "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
              "expect": {
```

```
    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
```

```
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
```

```
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

      "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,

     "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
```

```
disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
                      }
                    },
                    "params": {
                      "UseFIPS": false,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                    "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },
                  {
                    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                    "expect": {
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": true,
                      "Endpoint": "https://example.com"
                     }
                  },
                  {
                    "documentation": "Missing region",
                    "expect": {
                      "error": "Invalid Configuration: Missing Region"
                    }
                  }
                ],
                "version": "1.0"
              }
            }
          },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
      "type": "structure",
      "members": {
        "ProvisioningStatus": {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
          "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
          }
        },
        "HealthStatus": {
          "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
          "traits": {
            "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
          }
        },
        "Ipv4Address": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
          }
        },
        "Ipv6Address": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
```

```
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>"
          }
        },
        "ServerEndpoint": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
          "traits": {
            "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
          }
        },
        "LicenseServerEndpointId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
          }
        },
        "LicenseServerEndpointProvisioningStatus": {
          "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
          "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
          }
        },
        "LicenseServers": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
          }
        },
```

```
            "CreationTime": {
               "target": "smithy.api#Timestamp",

      "traits": {
                  "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
         "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
         "type": "string",
         "traits": {
            "smithy.api#enum": [
               {

               "name": "PROVISIONING",
                  "value": "PROVISIONING",
                  "documentation": ""
               },
               {
                  "name": "PROVISIONING_FAILED",
                  "value": "PROVISIONING_FAILED",
                  "documentation": ""
               },
               {
                  "name": "PROVISIONED",
                  "value": "PROVISIONED",
                  "documentation": ""
               },
               {
                  "name": "DELETING",
                  "value": "DELETING",
                  "documentation": ""
               },
```

```
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
              "name": "DELETED",

"value": "DELETED",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
              "name": "NOT_APPLICABLE",
              "value": "NOT_APPLICABLE",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
        "type": "list",
         "member": {
           "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }

  }
      },
      "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
         {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
             "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "IdentityProviderSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
             }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
```

```
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
         },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
       "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
```

```
        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
```

```
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits":
{
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
```

```
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
```

```
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

}
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListUserAssociations"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "InstanceUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
 {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product
```

```
user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
                }
            },
            "SubscriptionStartDate": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
            }
        },
        "SubscriptionEndDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
        "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
 reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
 license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
```

```
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/RegisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

          },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"

        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
      "type": "structure",
      "members": {
        "SecretId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
            "smithy.api#length": {
              "min": 1
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 5,
          "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
      "type": "structure",
      "members": {
        "Endpoint": {
```

```
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Settings": {
          "type": "structure",
          "members": {
            "Subnets": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
              "traits": {
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }

      },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
          "type": "operation",
          "input":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
```

```
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
```

```
            },
            "IdentityProvider":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
                 }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                    "smithy.api#required": {}
                }

     }
        },
        "traits": {
```

```
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {

    "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {

          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
```

```
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            },
            "TagKeys": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
           },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
          "smithy.api#http": {

    "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
            }
```

```
        },
        "UpdateSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
          }

        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
      "type": "structure",
      "members": {
        "AddSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",
            "smithy.api#required": {}
          }
        },
        "RemoveSubnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
```

```
                "smithy.api#required": {}
              }
            },
            "SecurityGroupId": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
              "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
              }
            }
          },

          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
          }
        }
      }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
```

```
 params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License
```

```
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
 != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
        region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
  ${{ matrix.python-version }}
       uses: actions/setup-python@v2
```

```
        with:
          python-version: ${{ matrix.python-version }}
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
```

```
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData
```

```
// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
```

```
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }
```

```
 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}
```

```
type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
 []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string
```

```
 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
```

```
nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
```

```
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
```

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

```
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {
```

```
 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
```

```
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
```

```
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
```

```
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
```

```
            },

        {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/DeregisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                 "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
```

```
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
             }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
          "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
```

```
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
```

```
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        },
```

```
            "Region": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
               }
            },
            "UsageOperation": {
               "target": "smithy.api#String",

   "traits": {
                  "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
               }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
            },
            "LastUpdatedTime": {
               "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
               }
            },
            "SubscriptionName": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
               }
            },
            "OsVersion": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
               }
            },
            "SubscriptionProviderCreateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp when you registered
```

```
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
                }
            },
            "SubscriptionProviderUpdateTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
                }
            },
            "DualSubscription": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
                }
            },
            "RegisteredWithSubscriptionProvider": {
                "target": "smithy.api#String",
                "traits": {

      "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
       },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
         "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
```

```
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {
        "Region": {
          "builtIn": "AWS::Region",
          "required": false,
          "documentation": "The AWS region used to dispatch the request.",
          "type": "String"
        },
        "UseDualStack": {
          "builtIn": "AWS::UseDualStack",
          "required": true,
          "default": false,
          "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
           "type": "Boolean"
        },
        "UseFIPS": {
          "builtIn": "AWS::UseFIPS",
          "required": true,
          "default": false,
          "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
          "type": "Boolean"
        },
        "Endpoint": {
          "builtIn": "SDK::Endpoint",
          "required": false,
```

```
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        }

],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
                                        },
                                        true
                                    ]
```

```
}
                        ],
                        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                        "type": "error"
                    },
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": {
                                "ref": "Endpoint"
                            },
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"

            }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "isSet",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {

            "fn": "aws.partition",
                                    "argv": [
                                        {
                                            "ref": "Region"
                                        }
                                    ],
```

```
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {

                    "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }

],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"

        },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    },
                    {
                        "fn": "booleanEquals",
```

```
                "argv": [
                  true,
                  {
                    "fn": "getAttr",
                    "argv": [

                     {
                          "ref": "PartitionResult"
                       },
                       "supportsDualStack"
                    ]
                  }
                ]
              }
            ],
            "rules": [
              {
                "conditions": [],

    "rules": [
                  {
                    "conditions": [],
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                      "properties": {},
                      "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],

            "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {
```

```
            "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {

          "fn": "booleanEquals",
                        "argv": [
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"
                              },
                              "supportsFIPS"
                            ]
                          },
                          true

                      ]
                        }
                      ],
                      "rules": [
                        {
                          "conditions": [],
                          "rules": [
                            {
                              "conditions": [],
                              "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                "headers": {}
                              },
                              "type": "endpoint"
                            }
                          ],
```

```
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled
but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true

]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {

                                "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "rules": [
```

```
                                        {
                                            "conditions": [],
                                            "rules": [

        {
                                                    "conditions": [],
                                                    "endpoint": {
                                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                        "properties": {},
                                                        "headers": {}
                                                    },
                                                    "type": "endpoint"
                                                }
                                            ],
                                            "type": "tree"

                                }
                                    ],
                                    "type": "tree"
                                },
                                {
                                    "conditions": [],
                                    "error": "DualStack is enabled but this partition does not support DualStack",
                                    "type": "error"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"
                        }
                    ],
                    "type": "tree"
```

```
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ],
        "type": "tree"
    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

  }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
```

```
                "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
              }

        },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {

    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
```

```
            }

        },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
```

```
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
```

```
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },

    {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
          }
```

```
                },
                {
                  "documentation": "Missing region",

                "expect": {
                      "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
              },
              {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
   <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
   <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
          <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
```

```
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the
returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
```

```
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
              "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
```

```
        "type": "structure",
        "members": {
          "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
```

```
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "RegisteredSubscriptionProviders"
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
          "type": "structure",
          "members": {
            "SubscriptionProviderSources": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

     "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
              }
            }
          },

          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
          "type": "structure",
          "members": {
            "RegisteredSubscriptionProviders": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
```

```
      "type": "structure",
      "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "tags": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "EQUAL",
            "value": "Equal",
            "documentation": "Equal operator"
          },
          {
            "name": "NOT_EQUAL",
            "value": "NotEqual",
            "documentation": "Not equal operator"
          },
          {
            "name": "CONTAINS",
            "value": "Contains",
```

```
            "documentation": "Contains operator"
          }
        ],
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

"name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {

             "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
          }
        },
        "SubscriptionProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
 subscription provider \n\t\t\tthat you requested.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
 Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
              }
            },
            "SubscriptionProviderStatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
              }
            },
            "LastSuccessfulDataRetrievalTime": {
              "target": "smithy.api#String",
              "traits": {

                "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
```

```
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "InProgress",
              "value": "InProgress",
              "documentation": "InProgress status"
             },
            {
              "name": "Completed",
              "value": "Completed",
              "documentation": "Completed status"
            },
            {
              "name": "Successful",
              "value": "Successful",
              "documentation": "Successful status"
            },
            {
              "name": "Failed",
              "value": "Failed",
              "documentation": "Failed status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#length": {
              "min": 1,
              "max": 100
```

```
            }
          }
        },
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          },
          "smithy.api#uniqueItems": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with
Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace.
Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing
from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
```

```
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

       "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
```

```
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
```

```
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
          "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",
```

```
    "members": {
        "resourceArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        },
        "tagKeys": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input":
{}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
```

```
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
          "type": "structure",
          "members": {
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
{}
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowUpdate": {
              "target": "smithy.api#Boolean",
              "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
              }
            }

          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
          "type": "structure",
          "members": {
```

```
            "LinuxSubscriptionsDiscovery": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
              "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
              }
            },
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
              }

    },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
              }
             }
          }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
```

```
        }
      }
    }
  }
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.93 sirupsen-logrus 1.9.3

1.93.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.94 aws-sdk-go-v2-service-s3 1.89.0

1.94.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
 // This member is required.
 Issuer *types.Issuer

 // License name.
 //
 // This member is required.
 LicenseName *string

 // Product name.
 //
 // This member is required.
 ProductName *string

 // Product SKU.
 //
```

```
 // This member is required.
 ProductSKU *string

 // Date and time range during which the license is valid, in ISO8601-UTC format.
 //
 // This member is required.
 Validity *types.DatetimeRange

 // Information about
 the license.
 LicenseMetadata []types.Metadata

 // Tags to add to the license. For more information about tagging support in
 // License Manager, see the [TagResource]operation.
 //
 // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
 Tags []types.Tag

 noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

 // Amazon Resource Name (ARN) of the license.
 LicenseArn *string

 // License status.
 Status types.LicenseStatus

 // License version.
 Version *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
```

```
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
```

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
```

```
//   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
```

```
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
```

```
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
```

```
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
```

```
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
          }

 },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

          {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}

  }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "LicenseServerEndpoint": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
              "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

         },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
```

```
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
```

```
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members":
{
            "Attribute": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
              }
            },
            "Operation": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
              }
            },
            "Value": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
            }

          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that
```

```
contains the user information for the product subscription.</p>"
                }
            },
            "AssociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
                }
            },
            "DisassociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
```

```
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
```

```
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
```

```
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
```

```
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },

"properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                 "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

```
true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

"fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      }
                    ]
                  },

                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
```

```
                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }

            ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [

        {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
```

```
                        "argv": [

                    {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "endpoint": {

    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS is enabled but this partition does not support FIPS",
        "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                "fn": "booleanEquals",
                "argv": [
                    {
                    "ref": "UseDualStack"
                    },
                    true
                ]
                }
            ],
            "rules": [
```

```
                {

            "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]

        }
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }

                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "DualStack is enabled but this partition does not support DualStack",
                  "type": "error"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
```

```
                    "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",

        "UseFIPS": true,
              "UseDualStack": false
            }
```

```
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

  "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
 true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
```

```
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
```

```
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
```

```
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

      "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,

    "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
```

```
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
          },
          "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
              "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
          },
          "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
          },
          "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>"
            }
          },
          "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
            }
          },
          "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
```

```
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
```

```
            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
              "name": "DELETED",

"value": "DELETED",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
```

```
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
              }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
                "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "IdentityProviderSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
         "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources that contain details about the Active Directory identity providers that meet the request criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum number of results
```

```
to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

     "members": {
            "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
     "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
```

```
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
        "members": {
          "MaxResults": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
 {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
```

```
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
```

```
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "ProductUserSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

},
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}

 }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
```

```
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
 {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
              }
            },
            "SubscriptionStartDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
              }
            },
            "SubscriptionEndDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
```

reference to the credentials provider that's used for RDS license server user administration. ",
"smithy.api#required": {}
}
}
},
"traits": {

"smithy.api#documentation": " Server settings that are specific to a Remote Desktop Services (RDS) license server. "
}
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
"type": "operation",
"input": {
"target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
},
"errors": [
{
"target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
"smithy.api#documentation": " Registers an identity provider for user-based subscriptions. ",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/identity-provider/RegisterIdentityProvider"

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
                }

            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
              "smithy.api#length": {
                "min": 1
              }
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
```

```
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }

    },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC. "
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " Starts a product subscription for a user with the specified identity provider. <note> Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing. For more information, see Viewing your monthly charges in the Amazon Web Services Billing User Guide. </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

```
 "uri": "/user/StartProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
 {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
            }
          }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
    "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "Tags": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
          "smithy.api#error": "client"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {

         "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

        }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
```

```
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

  "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
```

```
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
```

```
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}
```

```
type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if
```

```
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
```

```
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
```

```
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
```

```
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
```

```
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod
```

```
// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
```

```
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
 middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
 }
 return nil
}
```

```
func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
```

```
//
// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
```

```
c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
```

```
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
```

```
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
```

```
 OperationName: "DeleteLicense",
 }
}
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
```

```
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
```

```
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
                 }
            },
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
                }
            },
```

```
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
         "type": "operation",
         "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/subscription/GetServiceSettings"

          },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
         "type": "structure",
         "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
```

```
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
```

```
            "InstanceID": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
               }
            },
            "InstanceType": {
               "target": "smithy.api#String",
               "traits":
{
                  "smithy.api#documentation": "<p>The instance type of the resource.</p>"
               }
            },
            "AccountID": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of the instance.</p>"
               }
            },
            "Region": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
               }
            },
            "UsageOperation": {
               "target": "smithy.api#String",

   "traits": {
                  "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
               }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
```

```
        },
        "LastUpdatedTime": {
          "target": "smithy.api#String",
           "traits": {
             "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
          }
        },
        "SubscriptionName": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
          }
        },
        "OsVersion": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
          }
        },
        "SubscriptionProviderCreateTime": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
          }
        },
        "SubscriptionProviderUpdateTime": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
          }
        },
        "DualSubscription": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
          }
        },
        "RegisteredWithSubscriptionProvider": {
          "target": "smithy.api#String",
          "traits": {

     "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager Linux Subscriptions",
           "arnNamespace": "license-manager-linux-subscriptions",
          "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
```

```
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
               "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
```

```
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": {
                          "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"

              }
              ],
              "type": "tree"
            },
            {
```

```
"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {

"fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ],
            "assign": "PartitionResult"
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {

                "ref": "UseFIPS"
                  },
                  true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
```

```
                }

    ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"

                    },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [

                                     {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

        "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
```

```
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }
                                    ],

                            "type": "tree"
                                    }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {

            "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {

                        "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "fn": "getAttr",
                                                "argv": [
                                                    {
                                                        "ref": "PartitionResult"
                                                    },
```

```
                                "supportsFIPS"
                            ]
                        },
                        true

                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS is enabled
but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
```

```
                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {

                                            "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "rules": [

        {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"

                    }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
```

```
                                    "type": "error"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ],
    "type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

}
            },
            "params": {
```

```
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

    },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
       },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
```

```
"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                      }

                  },
                    "params": {
                      "Region": "us-iso-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                      }
                    },
                    "params": {
                      "Region": "us-iso-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
```

```
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },

        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
```

```
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",

            "expect": {
                "error": "Invalid Configuration: Missing Region"
              }
            }
          ],
          "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled LinuxSubscriptionsDiscovery"
          },
          {
            "name": "Disabled",
```

```
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
```

```
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
```

```
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
              "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
```

```
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
              }
            }
          },

          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
          "type": "structure",
          "members": {
            "RegisteredSubscriptionProviders": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
              "traits": {
                "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
          },
          "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
          ],
          "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "EQUAL",
              "value": "Equal",
              "documentation": "Equal operator"
             },
            {
              "name": "NOT_EQUAL",
              "value": "NotEqual",
              "documentation": "Not equal operator"
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
          "smithy.api#length": {
            "min": 1,
            "max": 20
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {

      "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
```

```
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
```

```
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
```

```
            "value": "Completed",
            "documentation": "Completed status"
          },
          {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
          },
          {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#length": {
            "min": 1,
            "max": 100
          }
        }
      },
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        },
        "smithy.api#uniqueItems": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

       "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
```

```
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "ACTIVE",
              "value": "ACTIVE",
              "documentation": "ACTIVE status"
            },
            {
              "name": "INVALID",
              "value": "INVALID",
              "documentation": "INVALID status"
            },
            {
              "name": "PENDING",
              "value": "PENDING",
              "documentation": "PENDING status"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
```

```
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
          "type": "structure",

      "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tagKeys": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input":
 {}
          }
        },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {

 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required":
{}
          }
        },
```

```
            "LinuxSubscriptionsDiscoverySettings": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
              "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>",
                "smithy.api#required": {}
              }
            },
            "AllowUpdate": {
              "target": "smithy.api#Boolean",
              "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
              }
            }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
            }
          }

      },
            "Status": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
              }
            },
            "StatusMessage": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
              "traits": {
```

```
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
              }
            },
            "HomeRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
          }
        }
      }
    }
    CheckInLicense
     Initialize stack step
      spanInitializeStart
      InterceptExecution
      RegisterServiceMetadata
      legacyEndpointContextSetter
      SetLogger
      OperationInputValidation
      spanInitializeEnd
     Serialize stack step
      spanBuildRequestStart
      setOperationInput
      ResolveEndpoint
      InterceptBeforeSerialization
      OperationSerializer
      InterceptAfterSerialization
     Build stack step
      ClientRequestID
      ComputeContentLength
      SetCredentialSourceMiddleware
      UserAgent
```

AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string
```

```
 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step

spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter

SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.95 common 0.67.2

1.95.1 Available under license :

Common libraries shared by Prometheus Go components.
Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.96 prometheus-common 0.67.2

1.96.1 Available under license :

Common libraries shared by Prometheus Go components.
Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.97 cjson 1.7.19-r0

1.97.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).
This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.97.2 Available under license :

Apache License 2.0
Contributors
==========

Original Author:
- [Dave Gamble](https://github.com/DaveGamble)

Current Maintainer:
- [Max Bruckner](https://github.com/FSMaxB)
- [Alan Wang](https://github.com/Alanscut)

Contributors:
* [Ajay Bhargav](https://github.com/ajaybhargav)
* [AlexanderVasiljev](https://github.com/AlexanderVasiljev)
* [Alper Akcan](https://github.com/alperakcan)
* [Andrew Tang](https://github.com/singku)
* [Andy](https://github.com/mlh0101)
* [Anton Sergeev](https://github.com/anton-sergeev)
* [Benbuck Nason](https://github.com/bnason-nf)
* [Bernt Johan Damslora](https://github.com/bjda)
* [Bob Kocisko](https://github.com/bobkocisko)
* [Christian Schulze](https://github.com/ChristianSch)
* [Casperinous](https://github.com/Casperinous)
* [ChenYuan](https://github.com/zjuchenyuan)
* [Debora Grosse](https://github.com/DeboraG)
* [dieyushi](https://github.com/dieyushi)
* [Dngwn Hung ()](https://github.com/DongwenHuang)
* [Dominik](https://github.com/DL6ER)
* [Donough Liu](https://github.com/ldm0)
*
[Erez Oxman](https://github.com/erez-o)
* Eswar Yaganti
* [Evan Todd](https://github.com/etodd)

* [Fabrice Fontaine](https://github.com/ffontaine)
* Ian Mobley
* Irwan Djadjadi
* [hopper-vul](https://github.com/hopper-vul)
* [HuKeping](https://github.com/HuKeping)
* [IvanVoid](https://github.com/npi3pak)
* [Jakub Wilk](https://github.com/jwilk)
* [Jiri Zouhar](https://github.com/loigu)
* [Jonathan Fether](https://github.com/jfether)
* [Joshua Arulsamy](https://github.com/jarulsamy)
* [Julian Ste](https://github.com/julian-st)
* [Julin Vsquez](https://github.com/juvasquezg)
* [Junbo Zheng](https://github.com/Junbo-Zheng)
* [Kevin Branigan](https://github.com/kbranigan)
* [Kevin Sapper](https://github.com/sappo)
* [Kyle Chisholm](https://github.com/ChisholmKyle)
* [Linus Wallgren](https://github.com/ecksun)
* [Luo Jin](https://github.com/Up-wind)
* [Max](https://github.com/maebex)
* [MaxBrandtner](https://github.com/MaxBrandtner)
* [Mateusz Szafoni](https://github.com/raiden00pl)
*
Mike Pontillo
* [miaoerduo](https://github.com/miaoerduo)
* [mohawk2](https://github.com/mohawk2)
* [Mike Jerris](https://github.com/mjerris)
* [Mike Robinson](https://github.com/mhrobinson)
* [Moorthy](https://github.com/moorthy-bs)
* [myd7349](https://github.com/myd7349)
* [NancyLi1013](https://github.com/NancyLi1013)
* [Orri](https://github.com/sbvoxel)
* Paulo Antonio Alvarez
* [Pawe Malowany](https://github.com/PawelMalowany)
* [Pawel Winogrodzki](https://github.com/PawelWMS)
* [prefetchnta](https://github.com/prefetchnta)
* [Rafael Leal Dias](https://github.com/rafaeldias)
* [Randy](https://github.com/randy408)
* [raiden00pl](https://github.com/raiden00pl)
* [Robin Mallinson](https://github.com/rmallins)
* [Rod Vagg](https://github.com/rvagg)
* [Roland Meertens](https://github.com/rmeertens)
* [Romain Porte](https://github.com/MicroJoe)
* [SANJEEV BA](https://github.com/basanjeev)
* [Sang-Heon Jeon](https://github.com/lntuition)
* [Sayan Bandyopadhyay](https://github.com/saynb)
*
[Simon Sobisch](https://github.com/GitMensch)
* [Simon Ricaldone](https://github.com/simon-p-r)

* [Stoian Ivanov](https://github.com/sdrsdr)
* [SuperH-0630](https://github.com/SuperH-0630)
* [Square789](https://github.com/Square789)
* [Stephan Gatzka](https://github.com/gatzka)
* [Tony Langhammer](https://github.com/BigBrainAFK)
* [Vemake](https://github.com/vemakereporter)
* [vwvw](https://github.com/vwvw)
* [warmsocks](https://github.com/warmsocks)
* [Wei Tan](https://github.com/tan-wei)
* [Weston Schmidt](https://github.com/schmidtw)
* [xiaomianhehe](https://github.com/xiaomianhehe)
* [yangfl](https://github.com/yangfl)
* [yuta-oxo](https://github.com/yuta-oxo)
* [Zach Hindes](https://github.com/zhindes)
* [Zhao Zhixu](https://github.com/zhaozhixu)
* [10km](https://github.com/10km)

And probably more people on [SourceForge](https://sourceforge.net/p/cjson/bugs/search/?q=status%3Aclosed-rejected+or+status%3Aclosed-out-of-date+or+status%3Awont-fix+or+status%3Aclosed-fixed+or+status%3Aclosed&page=0)

Also
thanks to all the people who reported bugs and suggested new features.
Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF
OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The MIT License (MIT)

Copyright (c) <year> 2007-14 Mike Karlesky, Mark VanderVoord, Greg Williams

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.98 prometheus-client-model 0.6.2

1.98.1 Available under license :

Data model artifacts for Prometheus.
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,

indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.99 prometheus-client 1.23.2

1.99.1 Available under license :

Copyright (c) 2013 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Prometheus instrumentation library for Go applications
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).

The following components are included in this product:

perks - a fork of https://github.com/bmizerany/perks
https://github.com/beorn7/perks
Copyright 2013-2015 Blake Mizerany, Bjrn Rabenstein
See https://github.com/beorn7/perks/blob/master/README.md for license details.

Go support for Protocol Buffers - Google's data interchange format
http://github.com/golang/protobuf/

Copyright 2010 The Go Authors
See source code for license details.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.100 xxhash 2.3.0

1.100.1 Available under license :

Copyright (c) 2016 Caleb Spare

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.101 procfs 0.19.2

1.101.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
procfs provides functions to retrieve system, kernel and process
metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).

1.102 aws-sdk-go-v2-feature-s3-manager 1.20.1

1.102.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput

```
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement
```

```
// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string
```

```
// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
```

```
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err
 = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
```

```
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
```

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
```

```
 return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
  - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
```

```
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
```

```
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
```

```
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
```

```
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
```

```
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
```

```
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
```

AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
```

```
        "members": {
          "DirectoryId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
            "traits": {
              "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
            }
          },
          "ActiveDirectorySettings":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
```

```
                "smithy.api#documentation": "<p>A list of domain IPv4
 addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainIpv6List": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
              "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 2
                }
              }
            },
            "DomainCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
              }
            },
            "DomainNetworkSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
          "type": "string",
          "traits": {
            "smithy.api#enum": [
              {
                "name": "SELF_MANAGED",
                 "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
              },
```

```
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

                "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
                }
```

```
            },
            "Tags": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
               "traits": {
                  "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
         "type": "structure",
         "members": {
            "InstanceUserSummary": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
         "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
            "smithy.api#error": "server"
         }

   },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
```

```
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

        {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
```

Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details. ",
"smithy.api#required": {}
}
},
"LicenseServerSettings": {
"target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
"traits": {
"smithy.api#documentation": " The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server. ",
"smithy.api#required": {}
}
},
"Tags": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
"traits": {
"smithy.api#documentation": " The tags that apply for the license server endpoint. "
}
}
},
"traits": {
"smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
"type": "structure",
"members": {
"IdentityProviderArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The Amazon Resource Name (ARN) of the identity
provider specified in the request. "
}
},
"LicenseServerEndpointArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The ARN of the <code>LicenseServerEndpoint</code> resource. "
}
}
},
"traits": {
"smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {

```
        "type": "union",
        "members": {
          "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
```

```
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                    "smithy.api#required": {}

  }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
```

```
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

   },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

    },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
```

```
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
```

```
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
          "type": "operation",
          "input": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

   "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
```

```
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
```

```
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
          }
        },

    "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
{
            "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
            },
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC
```

endpoints. ",

"smithy.api#required": {}
}
},
"Product": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the user-based subscription product. ",
"smithy.api#required": {}
}
},
"Status": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The status of the identity provider. ",
"smithy.api#required": {}
}
},
"IdentityProviderArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The Amazon Resource Name (ARN) of the identity provider. "
}

},
"FailureMessage": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The failure message associated with an identity provider. "
}
},
"OwnerAccountId": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The AWS Account ID of the owner of this resource. "
}
}
},
"traits": {
"smithy.api#documentation": " Describes an identity provider. "
}
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
"type": "list",
"member": {
"target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
}

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
            },
            "OwnerAccountId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
```

```
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
```

```
                "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
              }
            },
            "InstanceUserArn": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
       }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
```

```
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager User Subscriptions",
          "arnNamespace": "license-manager-user-subscriptions",
          "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
             "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
```

```
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]

    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
```

```
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [

{
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },

        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
```

```
                {
                  "fn": "aws.partition",
                   "argv": [
                    {
                      "ref": "Region"
                    }
                  ],
                  "assign": "PartitionResult"
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {

"fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
```

```
                                    ]
                                }
                            ]
                        },

                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }

                    ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }

            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [
```

```
                {
                  "fn": "booleanEquals",
                  "argv": [

{
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "fn": "getAttr",
                          "argv": [

                {
                              "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                          ]
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {

          "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "FIPS is enabled but this partition does not support FIPS",
```

```
                        "type": "error"

            }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ],
                    "rules": [
                        {

        "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        true,
                                        {
                                            "fn": "getAttr",
                                            "argv": [
                                                {
                                                    "ref": "PartitionResult"
                                                },
                                                "supportsDualStack"
                                            ]

         }
                                    ]
                                }
                            ],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
```

```
                                    "type": "endpoint"
                                }

                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "error": "DualStack is enabled but this partition does not support DualStack",
                            "type": "error"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                        "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                }
            },
```

```
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",

  "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
```

```
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS":
true,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                  }
                },
                "params": {
                  "Region": "cn-north-1",
                  "UseFIPS": false,
                  "UseDualStack": true
                }
              },
              {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                  }
                },
```

```
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {

"endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
```

```
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
```

```
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": true,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                "expect": {

        "endpoint": {
                      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                  }
                },
                "params": {
                  "Region": "us-isob-east-1",
                  "UseFIPS": false,
                  "UseDualStack": false
                }
              },
              {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,

         "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                  "endpoint": {
                    "url": "https://example.com"
                  }
                },
                "params": {
                  "UseFIPS": false,
                  "UseDualStack": false,
                  "Endpoint": "https://example.com"
                }
```

```
                },
                {
                  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                  "expect": {

 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                   }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
          "type": "structure",
          "members": {
            "ProvisioningStatus": {
              "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
              }
            },
```

```
            "HealthStatus": {
               "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
               "traits": {
                  "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
               }
            },
            "Ipv4Address": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
               }
            },
            "Ipv6Address": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
          }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
         "type": "structure",
         "members": {
            "IdentityProviderArn": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
               }
            },
            "ServerType": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
               "traits": {
                  "smithy.api#documentation": "<p>The type of license server.</p>"
               }
            },
            "ServerEndpoint": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
               "traits": {
                  "smithy.api#documentation": "<p>The
```

<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint. "
}
},
"StatusMessage": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The message associated with the provisioning status, if there is one. "
}
},
"LicenseServerEndpointId": {
"target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
"traits": {
"smithy.api#documentation": " The ID of the license server endpoint. "
}
},
"LicenseServerEndpointArn": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
"traits": {
"smithy.api#documentation": " The ARN of the <code>ServerEndpoint</code> resource for the RDS license server. "
}
},
"LicenseServerEndpointProvisioningStatus": {
"target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
"traits": {
"smithy.api#documentation": " The current state of the provisioning process for the RDS license server endpoint "
}
},
"LicenseServers": {
"target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
"traits": {
"smithy.api#documentation": " An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint. "
}
},
"CreationTime": {
"target": "smithy.api#Timestamp",

"traits": {
"smithy.api#documentation": " The timestamp when License Manager created the license server endpoint. "
}
}
},
"traits": {

```
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {

                "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
                    "value": "PROVISIONING_FAILED",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONED",
                    "value": "PROVISIONED",
                    "documentation": ""
                },
                {
                    "name": "DELETING",
                    "value": "DELETING",
                    "documentation": ""
                },
                {
                    "name": "DELETION_FAILED",
                    "value": "DELETION_FAILED",
                    "documentation": ""
                },
                {
                    "name": "DELETED",

 "value": "DELETED",
                    "documentation": ""
```

```
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
              "name": "NOT_APPLICABLE",
              "value": "NOT_APPLICABLE",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
        "type": "list",
         "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
        "type": "structure",
        "members": {
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>",
              "smithy.api#required": {}
            }
          },
          "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
              "smithy.api#required": {}
```

```
            }

    }
        },
        "traits": {
            "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
```

```
                "inputToken": "NextToken",
                 "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "IdentityProviderSummaries"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
        "type": "structure",
        "members": {
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
                 }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
 criteria.</p>",
                    "smithy.api#required": {}
                }
            },
```

```
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
```

```
            "method": "POST",
            "uri": "/instance/ListInstances"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
        "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
 from a previously truncated response.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
 <li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

       "members": {
          "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
```

contain details about the instances that provide user-based subscriptions and also meet the request criteria. "
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": " The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects. "
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

      {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " List the Remote Desktop Services (RDS) License Server endpoints ",

```
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "LicenseServerEndpoints"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

   "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                  "smithy.api#range": {
                     "min": 1,
                     "max": 100
                  }
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
               "traits": {
                  "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
               }
            },
            "NextToken": {
               "target": "smithy.api#String",
               "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#input": {}
         }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoints": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },

        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
               "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "ProductUserSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
         "type": "structure",
         "members": {
            "Product": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                  "smithy.api#required": {}
               }
            },
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
```

```
request.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

  },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
  you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}

}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/user/ListUserAssociations"
```

```
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "InstanceUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
          "type": "structure",
          "members": {
            "InstanceId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

      "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
              }
```

```
            }
          },
          "traits": {
            "smithy.api#input": {}
          }

    },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
          "type": "structure",
          "members": {
            "InstanceUserSummaries": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output":
    {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                "smithy.api#required": {}
              }
            },
```

```
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
```

```
        "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {

            "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
          "type": "structure",
          "members": {
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }

            },
            "Settings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
              "traits": {
```

```
                "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {

         "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
```

```
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
      "type": "union",

     "members": {
        "RdsSalSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
```

```
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                  "min": 1
                },
                "smithy.api#required": {}
              }

    },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
            "code": 200,
            "method": "POST",

 "uri": "/user/StartProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
            },
            "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
            },
            "Product": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
```

```
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
             },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/StopProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
              }
            },
            "IdentityProvider":
{
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
              }
            },
            "Product": {
              "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
              "smithy.api#required": {}
            }

      }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
```

```
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {

   "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
        "smithy.api#http": {
```

```
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

             "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
```

```
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

          }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
            "smithy.api#http": {

    "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn":
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
```

```
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
          }
```

```
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A parameter is not valid.</p>",
          "smithy.api#error": "client"
        }
      }
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
```

```
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options);
 err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
```

```
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options);
 err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan
```

```
on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
     - name: Checkout this ref
       uses: actions/checkout@v2
       with:
         path: new-ref
         fetch-depth: 0
     - name: Get Diff
       run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
     - name: Get Target Files
       run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
     - name: Checkout scancode
       uses: actions/checkout@v2
       with:
         repository: nexB/scancode-toolkit
         path: scancode-toolkit
         fetch-depth: 1
     - name: Set up Python
 ${{ matrix.python-version }}
       uses: actions/setup-python@v2
       with:
         python-version: ${{ matrix.python-version }}
     # ScanCode
     - name: Self-configure scancode
       working-directory: ./scancode-toolkit
       run: ./scancode --help
     - name: Run Scan code on target
       run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
```

```
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType
```

```
 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
```

```
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
```

```
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
```

```
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
```

```
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

 // Information about constraints.
 CheckoutMetadata []types.Metadata

 // Allowed license entitlements.
 EntitlementsAllowed
```

```
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
```

```
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return
 err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
```

```
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
```

```
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
```

```
 return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
 middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
```

```
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
```

```
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string
```

```
 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
```

```
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err !=
 nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
{
  "smithy": "2.0",
  "shapes": {
     "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
     },
     "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
     },
     "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
           "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
           "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
           },
           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
           },
```

```
    {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/DeregisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
              }
            },
            "Values": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
              }
            },
            "Operator": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
              "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
          "type":
 "list",
          "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
          },
          "output": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
             },
            {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
                }
            },
```

```
            "SecretArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
               }
            },
            "SubscriptionProviderStatus": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
               "traits": {
                  "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
                }
            },
            "SubscriptionProviderStatusMessage": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
               }
            },
            "LastSuccessfulDataRetrievalTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
         "type": "operation",
         "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
         },
         "errors": [
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"

            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
                }
            },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
```

```
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                 "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
            "AmiId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
                }
            },
            "InstanceID": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
                }
            },
            "InstanceType": {
                "target": "smithy.api#String",
                "traits":
{
                    "smithy.api#documentation": "<p>The instance type of the resource.</p>"
                }
            },
            "AccountID": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the instance.</p>"
                }
            },
            "Region": {
                "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
            }
        },
        "UsageOperation": {
            "target": "smithy.api#String",

   "traits": {
                "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
            }
        },
        "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
```

```
            },
            "SubscriptionProviderUpdateTime": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
               }
            },
            "DualSubscription": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
               }
            },
            "RegisteredWithSubscriptionProvider": {
               "target": "smithy.api#String",
               "traits": {

     "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
         "type": "list",
         "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
         "type": "structure",
         "members": {
            "message": {
               "target": "smithy.api#String"
            }
         },
         "traits": {
            "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server"
         }
      },
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
         },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ],
    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager Linux Subscriptions",
             "arnNamespace": "license-manager-linux-subscriptions",
            "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
```

```
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
               "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
```

```
        }
      },
      "rules": [
        {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Endpoint"
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
```

```
                "type": "error"
            },
            {
                "conditions": [],
                "endpoint": {
                    "url": {
                        "ref": "Endpoint"
                    },
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {

    "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
```

```
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {

                                "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            },
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }

],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            true,
                                            {
                                                "fn": "getAttr",
                                                "argv": [
                                                    {
                                                        "ref": "PartitionResult"

            },
                                                        "supportsFIPS"
                                                    ]
                                                }
                                            ]
                                        },
                                        {
                                            "fn": "booleanEquals",
                                            "argv": [
                                                true,
                                                {
```

```
                                        "fn": "getAttr",
                                        "argv": [

                                      {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                      ]
                                    }
                                  ]
                                }
                              ],
                              "rules": [
                                {
                                  "conditions": [],

      "rules": [
                                    {
                                      "conditions": [],
                                      "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                      },
                                      "type": "endpoint"
                                    }
                                  ],

                          "type": "tree"
                                  }
                                ],
                                "type": "tree"
                              },
                              {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                                "type": "error"
                              }
                            ],
                            "type": "tree"
                          },
                          {

          "conditions": [
                              {
                                "fn": "booleanEquals",
```

```
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

        "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true

                  ]
                  }
                ],
                "rules": [
                  {
                    "conditions": [],
                    "rules": [
                      {
                        "conditions": [],
                        "endpoint": {
                          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                          "headers": {}
                        },
                        "type": "endpoint"
                      }
                    ],
                    "type": "tree"
                  }
                ],
```

```
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled
but this partition does not support FIPS",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true

]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {

                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "rules": [
```

```
        {
                                                "conditions": [],
                                                "endpoint": {
                                                   "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                   "properties": {},
                                                   "headers": {}
                                                },
                                                "type": "endpoint"
                                             }
                                          ],
                                          "type": "tree"

                                    }
                                       ],
                                       "type": "tree"
                                    },
                                    {
                                       "conditions": [],
                                       "error": "DualStack is enabled but this partition does not support DualStack",
                                       "type": "error"
                                    }
                                 ],
                                 "type": "tree"
                              },
                              {
                                 "conditions": [],
                                 "rules": [
                                    {
                                       "conditions": [],
                                       "endpoint": {
                                          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                          "properties": {},
                                          "headers": {}
                                       },
                                       "type": "endpoint"
                                    }
                                 ],
                                 "type": "tree"
                              }
                           ],
                           "type": "tree"
                        }
                     ],
                     "type": "tree"
                  },
```

```
            {
              "conditions": [],
              "error": "Invalid Configuration: Missing Region",
              "type": "error"
            }
          ],
          "type": "tree"
        }
      ]
    },
    "smithy.rules#endpointTests": {
      "testCases": [
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

      }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

       "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
```

```
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

      },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": false,
              "UseDualStack": false
            }
          },
          {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
              }
            },
            "params": {
              "Region": "cn-north-1",
              "UseFIPS": true,
              "UseDualStack": false
            }
          },
          {
```

```
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }

      },
      "params": {
```

```
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }

        },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
```

```
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

}
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
```

```
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },

        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",
```

```
                "expect": {
                    "error": "Invalid Configuration: Missing Region"
                }
                }
            ],
            "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled LinuxSubscriptionsDiscovery"
          },
          {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
      "type": "structure",
      "members": {
        "SourceRegions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
          "traits": {
            "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "OrganizationIntegration": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
          "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
```

```
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
            "Filters": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of structures
```

```
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
          "type": "structure",
          "members": {
            "Instances": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
              "traits": {
```

```
                "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional
 objects.</p>"
              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/ListLinuxSubscriptions"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "Subscriptions"
```

```
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Filters": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                  "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
                }
              },
              "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                  "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
                }
              },
              "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                  "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
                  "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                  }
                }
              }
            },
            "traits": {
              "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
            }
          },
          "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
            "type": "structure",
            "members": {
              "Subscriptions": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"

            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
```

```
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "RegisteredSubscriptionProviders"
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSources": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

  "traits": {
                    "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
                }
            },
            "MaxResults": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
                "traits": {
                    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
                    "smithy.api#range": {
                        "min": 1,
                        "max": 100
                    }
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
            "RegisteredSubscriptionProviders": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
                "traits": {
                    "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
```

```
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
           }
        ],
        "traits": {
          "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon
Web Services resource.</p>",
          "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#readonly": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}

            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "EQUAL",
              "value": "Equal",
              "documentation": "Equal operator"
            },
            {
              "name": "NOT_EQUAL",
              "value": "NotEqual",
              "documentation": "Not equal operator"
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
```

```
        "smithy.api#length": {
          "min": 1,
          "max": 20
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled OrganizationIntegration"
          },
          {

    "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled OrganizationIntegration"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },

       {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {

                "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
```

```
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
              }
            },
            "SubscriptionProviderArn": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
              }
            },
            "SubscriptionProviderStatus": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
              "traits": {
                "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
          "type": "structure",
          "members": {
            "SubscriptionProviderArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
              }
            },
            "SubscriptionProviderSource": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
              "traits": {
                "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
              }
            },
            "SecretArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
```

```
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
            }
          },
          "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
```

```
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#length": {
                    "min": 1,
                    "max": 100
                }
            }
```

```
        },
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            },
            "smithy.api#uniqueItems": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
        "type": "map",
        "key": {
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
            "Name": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the subscription.</p>"
                }
            },
            "Type": {
                "target": "smithy.api#String",
                "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with
Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace.
Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing
from Amazon EC2 or BYOS.</p>"
                }
            },
            "InstanceCount": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
                "traits": {
                    "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
    },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

     "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-
9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
```

```
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
              "method": "PUT",
              "uri": "/tags/{resourceArn}"
            },
```

```
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            },
            "tags": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\tformatted as key value pairs.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
```

```
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

  "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input":
{}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
```

```
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }

    },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
         }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
      }
    }
  }
```

}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {

```
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
```

```
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
```

spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.103 github.com/aws/aws-sdk-go-v2/service/s3 1.89.0

1.103.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware

```
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
```

```
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
 // This member is required.
 Issuer *types.Issuer
```

```
// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return
err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
```

```
 ServiceID:     ServiceID,
 OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
 spanInitializeStart
 InterceptExecution
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 spanInitializeEnd
 Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
```

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

```
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {
```

```
 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
 //   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
```

```
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
```

```
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
```

```
 spanInitializeEnd
Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
```

```
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
```

```
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
 err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
```

```
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
```

```
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
```

```
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
              "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
```

```
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
```

```
            "uri": "/user/AssociateUser"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
             "traits": {
               "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
               "smithy.api#required": {}
             }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
          "smithy.api#error": "server"
        }

    },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

         {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
         "type": "structure",
         "members":
{
            "IdentityProviderArn": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                  "smithy.api#required": {}
               }
            },
            "LicenseServerSettings": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
               "traits": {
                  "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
```

```
secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
                }
            },
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
                }
            }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
      "type": "structure",
      "members": {
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
            "smithy.api#required": {}

  }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoint": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
          "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

     },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/DeregisterIdentityProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
                }
            },
            "InstanceId": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based subscriptions.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members":
{
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Settings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
```

```
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
            }

          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
```

```
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        },
        "Products": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                "smithy.api#required": {}
            }
        },
        "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
        },
        "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
},
```

```
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {

            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
          }
        },
```

```
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
 contains the user information for the product subscription.</p>"
              }
            },
            "AssociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
              }
            },
            "DisassociationDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
 subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
          "type": "list",
          "member": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
```

```
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
        },
        {

            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
        }
    ],
    "traits": {
        "aws.api#service": {
            "sdkId": "License Manager User Subscriptions",
            "arnNamespace": "license-manager-user-subscriptions",
            "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
            "name": "license-manager-user-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
```

```
            "additionalAllowedHeaders": [
              "*",
              "content-type",
              "x-amz-content-sha256",
              "x-amz-user-agent",
              "x-amzn-platform-id",
              "x-amzn-trace-id",
              "content-length",
               "x-api-key",
              "authorization",
              "x-amz-date",
              "x-amz-security-token",
              "Access-Control-Allow-Headers",
              "Access-Control-Allow-Methods",
              "Access-Control-Allow-Origin"
            ],
            "additionalExposedHeaders": [
              "x-amzn-errortype",
              "x-amzn-requestid",
              "x-amzn-trace-id"
            ]
          },
          "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
          "smithy.api#title": "AWS License Manager User Subscriptions",
          "smithy.rules#endpointRuleSet": {
            "version": "1.0",
            "parameters": {

    "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
              },
              "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
              },
              "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
```

the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",


```
            "type": "boolean"
          },
          "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
          }
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

  }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
```

```
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],
        "endpoint": {
          "url": {
            "ref": "Endpoint"
          },

"properties": {},
            "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
},
{
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "aws.partition",
             "argv": [
               {
                  "ref": "Region"
               }
            ],
            "assign": "PartitionResult"
          }
        ],
        "rules": [
```

```
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseFIPS"
                        },

true
                      ]
                    },
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
                          "ref": "UseDualStack"
                        },
                        true
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [
                        {

"fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
                              "argv": [
                                {
                                  "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                              ]
                            }
                          ]
                        },

                        {
                          "fn": "booleanEquals",
                          "argv": [
                            true,
                            {
                              "fn": "getAttr",
```

```
                                    "argv": [
                                        {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }

                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
                        }

                ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                    "type": "error"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [

                {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
```

```
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "fn": "getAttr",
                                        "argv": [

                    {
                                                "ref": "PartitionResult"
                                            },
                                            "supportsFIPS"
                                        ]
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "FIPS is enabled but this partition does not support FIPS",
                        "type": "error"

}
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
```

```
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ],
          "rules": [
            {

      "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    true,
                    {
                      "fn": "getAttr",
                      "argv": [
                        {
                          "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                      ]

       }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }

              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "DualStack is enabled but this partition does not support DualStack",
              "type": "error"
            }
```

```
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                                "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "Invalid Configuration: Missing Region",
                "type": "error"
            }
        ]
    },
    "smithy.rules#endpointTests": {
        "testCases": [
            {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
```

```
            }
          },
          "params": {
            "Region": "us-east-1",

        "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

    "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
```

```
true,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {

"endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS":
false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                "expect": {
```

```
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
"endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
```

```
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint":
{
              "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
```

```
      "endpoint": {
                        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
                    }
                },
                "params": {
                    "Region": "us-isob-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    }
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,

      "UseDualStack": false,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                    "expect": {
                        "endpoint": {
                            "url": "https://example.com"
                        }
                    },
                    "params": {
                        "UseFIPS": false,
                        "UseDualStack": false,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
                    "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
                    },
                    "params": {
                        "Region": "us-east-1",
                        "UseFIPS": true,
```

```
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                  }
                },
                {
                  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
                  "expect": {
                    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
                  },
                  "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": true,
                    "Endpoint": "https://example.com"
                   }
                },
                {
                  "documentation": "Missing region",
                  "expect": {
                    "error": "Invalid Configuration: Missing Region"
                  }
                }
              ],
              "version": "1.0"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
          "type": "structure",
          "members": {
            "ProvisioningStatus": {
              "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
              "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
              }
            },
            "HealthStatus": {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
              "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
              }
            },
            "Ipv4Address": {
              "target": "smithy.api#String",
              "traits": {
```

```
            "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license server.</p>"
          }
        },
        "Ipv6Address": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license server.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>"
          }
        },
        "ServerEndpoint": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
          "traits": {
            "smithy.api#documentation": "<p>The
 <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
          }
        },
        "LicenseServerEndpointId": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
 resource for the RDS license server.</p>"
            }
          },
          "LicenseServerEndpointProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
            }
          },
          "LicenseServers": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
            }
          },
          "CreationTime": {
            "target": "smithy.api#Timestamp",

    "traits": {
              "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {

            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
              "name": "DELETED",

    "value": "DELETED",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
```

```
                "documentation": ""
              },
              {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
              },
              {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
              }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
               "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "IdentityProviderSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
          "type": "structure",
          "members": {
            "MaxResults": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources that contain details about the Active Directory identity providers that meet the request criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
```

```
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
     },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
       }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
      }
    }
  },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum number of results
 to return from a single request.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
          }
        },
        "Filters": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
      "type": "structure",

     "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
```

```
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },

  {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
      }
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },

          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
```

```
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
              "inputToken": "NextToken",
              "outputToken": "NextToken",
              "pageSize": "MaxResults",
              "items": "ProductUserSummaries"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
          "type": "structure",
          "members": {
            "Product": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
              }
            },
            "Filters": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
              "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
              }
            },
            "NextToken": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

  },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
```

```
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}

 }
    },
```

```
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
        }
    }
},
 "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
```

```
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
 {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
```

```
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
          }
        },
        "SubscriptionStartDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
          }
        },
        "SubscriptionEndDate": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
```

```
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {

        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS) license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
         {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
```

```
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/RegisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
                }

         },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
                }
            }
        },
        "traits": {
```

```
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
          "type": "structure",
          "members": {
            "IdentityProviderSummary": {

           "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                  "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                  "smithy.api#required": {}
                }
              }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
          "type": "string",
          "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"

            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
          "type": "structure",
          "members": {
            "SecretId": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
```

```
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
```

```
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }

    },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
```

communication between resources in your VPC and the VPC endpoint for activation servers. ",
"smithy.api#required": {}
}
}
},
"traits": {
"smithy.api#documentation": " The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC. "
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
"type": "operation",
"input":
{
"target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
{
"target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
},

{
"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
"smithy.api#documentation": " Starts a product subscription for a user with the specified identity provider. <note> Your estimated bill for charges on the number of users and related costs will take 48 hours

to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing. For more information, see Viewing your monthly charges in the Amazon Web Services Billing User Guide. </note>",
"smithy.api#http": {
"code": 200,
"method": "POST",

"uri": "/user/StartProductSubscription"
}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
"type": "structure",
"members": {
"Username": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The user name from the identity provider of the user. ",
"smithy.api#required": {}
}
},
"IdentityProvider": {
"target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
"traits": {
"smithy.api#documentation": " An object that specifies details for the identity provider. ",
"smithy.api#required": {}
}
},
"Product": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The name of the user-based subscription product. Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> ",
"smithy.api#required": {}
}
},
"Domain": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The domain name of the Active Directory that contains the user for whom to start the product subscription. "
}
},
"Tags": {
"target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
"traits": {

```
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
          "type": "structure",
          "members": {
            "ProductUserSummary": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
              "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
```

```
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/StopProductSubscription"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
                }
            },
```

```
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
            "smithy.api#required": {}
          }

     }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
```

```
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {

      "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
  "<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Tags": {
          "type": "map",
          "key": {
            "target": "smithy.api#String"
          },
          "value": {
            "target": "smithy.api#String"
          },
          "traits": {
            "smithy.api#length": {
              "min": 0,
              "max": 50
            },
            "smithy.api#sensitive": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {

                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

                }
            },
            "TagKeys": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
            "traits": {
              "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
              "smithy.api#httpQuery": "tagKeys",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
         },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
          "smithy.api#http": {

    "code": 200,
            "method": "POST",
```

```
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
          "AddSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
              "smithy.api#required": {}
            }
          },
          "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
              "smithy.api#required": {}
            }
          },
          "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
              "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
          }
        },

       "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
```

```
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
```

```
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
```

```
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
```

```
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
```

```
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }
```

```
if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
```

```
 err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack,
options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
```

```
      python-version: [3.9]

  steps:
    - name: Checkout target
      uses: actions/checkout@v2
      with:
        path: sdkbase
        ref: ${{ github.base_ref }}
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
```

```
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData
```

```
// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata
```

```
 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack,
 options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
```

```
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack,
options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
```

```
 Region:       region,
 ServiceID:    ServiceID,
 OperationName: "CheckoutLicense",
 }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string
```

```
// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string
```

```
// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
```

```
 return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }
```

```
 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
 GrafanaToken *string

 noSmithyDocumentSerde
}

type AssociateLicenseOutput
 struct {

 // A structure containing data about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string
```

```
 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
```

```
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
```

```
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
```

```
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "DeleteLicense",
 }
}
{
  "smithy": "2.0",
  "shapes": {
     "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
     },
     "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
     },
     "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
           "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
           "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
           },
           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
           },

     {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
           },
           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
           }
        ],
        "traits": {
           "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
           "smithy.api#http": {
              "code": 200,
```

```
            "method": "POST",
            "uri": "/subscription/DeregisterSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
 "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
           },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
          },
          "smithy.api#idempotent": {}
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

       "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
```

```
                }
            },
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
                }
            },
            "LastSuccessfulDataRetrievalTime": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
        "type": "operation",
        "input": {
            "target":
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetServiceSettings"
```

```
        },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
      "type": "structure",
      "members": {
        "AmiId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
          }
        },
        "InstanceID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
          }
        },
        "InstanceType": {
          "target": "smithy.api#String",
          "traits":
{
            "smithy.api#documentation": "<p>The instance type of the resource.</p>"
          }
        },
        "AccountID": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of the instance.</p>"
          }
        },
        "Region": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
          }
        },
        "UsageOperation": {
          "target": "smithy.api#String",

   "traits": {
            "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
```

```
            }
        },
        "ProductCode": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
            "traits": {
                "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
            }
        },
        "LastUpdatedTime": {
            "target": "smithy.api#String",
             "traits": {
                "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
            }
        },
        "SubscriptionName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
            }
        },
        "OsVersion": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
            }
        },
        "SubscriptionProviderCreateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
            }
        },
        "SubscriptionProviderUpdateTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
            }
        },
        "DualSubscription": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
```

\n\t\t\tthe same software on your instance. "
            }
          },
          "RegisteredWithSubscriptionProvider": {
            "target": "smithy.api#String",
            "traits": {

    "smithy.api#documentation": " Indicates that your instance uses a BYOL license subscription from \n\t\t\ta third-party Linux subscription provider that you've registered with License Manager. "
            }
          }
        },
        "traits": {
          "smithy.api#documentation": " Details discovered information about a running instance using Linux subscriptions. "
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 " An exception occurred with the service. ",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"

```
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
```

```
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
 you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
 support dual-stack, dispatching the request MAY return an error.",
               "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
 configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
```

```
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }

    ],
                    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                    "type": "error"
                },
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseDualStack"
                                        },
                                        true
                                    ]

}
                            ],
                            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                            "type": "error"
                        },
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": {
                                    "ref": "Endpoint"
                                },
                                "properties": {},
                                "headers": {}
                            },
                            "type": "endpoint"
```

```
            }
          ],
          "type": "tree"

    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
      {
          "conditions": [
            {
              "fn": "isSet",
              "argv": [
                {
                  "ref": "Region"
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [
                {

    "fn": "aws.partition",
                    "argv": [
                      {
                        "ref": "Region"
                      }
                    ],
                    "assign": "PartitionResult"
                  }
                ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          {

                      "ref": "UseFIPS"
                          },
                          true
                        ]
```

```
                },
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseDualStack"
                    },
                    true
                  ]
                }

    ],
                "rules": [
                  {
                    "conditions": [
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [
                              {
                                "ref": "PartitionResult"

    },
                                "supportsFIPS"
                            ]
                          }
                        ]
                      },
                      {
                        "fn": "booleanEquals",
                        "argv": [
                          true,
                          {
                            "fn": "getAttr",
                            "argv": [

                             {
                                "ref": "PartitionResult"
                              },
                              "supportsDualStack"
                            ]
                          }
                        ]
                      }
                    ],
```

```
                "rules": [
                  {
                    "conditions": [],

        "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],

                "type": "tree"
                  }
                ],
                "type": "tree"
              },
              {
                "conditions": [],
                "error": "FIPS and DualStack are enabled, but this partition does not support one or both",
                "type": "error"
              }
            ],
            "type": "tree"
          },
          {

      "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
```

```
                "fn": "booleanEquals",
                    "argv": [
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      },
                      true

                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "error": "FIPS is enabled
but this partition does not support FIPS",
              "type": "error"
            }
          ],
          "type": "tree"
        },
        {
```

```
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true

                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {

                                            "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "rules": [

        {
                                            "conditions": [],
                                            "endpoint": {
                                                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }
                                    ],
```

```
                                    "type": "tree"

                            }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "DualStack is enabled but this partition does not support DualStack",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                    "properties": {},
                                    "headers": {}
                                },
                                "type": "endpoint"
                            }
                        ],
                        "type": "tree"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
"testCases": [
```

```
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

}
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {

   "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
```

```
    },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": true
        }
      },
      {

  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
          }
        },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": true,
          "UseDualStack": false
        }
      },
      {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
          }
         },
        "params": {
          "Region": "cn-north-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
```

```
                {
                  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                  },
                  "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": false
                  }
                },
                {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
                    }
                  },
                  "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                  }
                },
                {
                  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
                    }

                 },
                  "params": {
                    "Region": "us-gov-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                  }
                },
                {
                  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
                  "expect": {
                    "endpoint": {
                      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
                    }
                  },
```

```
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }

    },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
```

```
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

  }
                    },
                    "params": {
                      "Region": "us-isob-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
                      }
                    },
                    "params": {
                      "Region": "us-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false,
                      "Endpoint": "https://example.com"
                    }
                  },

          {
                    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://example.com"
```

```
                }
              },
              "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",

            "expect": {
                "error": "Invalid Configuration: Missing Region"
              }
            }
          ],
          "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
      "type": "string",
      "traits": {
```

```
            "smithy.api#enum": [
               {
                  "name": "Enabled",
                  "value": "Enabled",
                  "documentation": "Enabled LinuxSubscriptionsDiscovery"
               },
               {
                  "name": "Disabled",
                  "value": "Disabled",
                  "documentation": "Disabled LinuxSubscriptionsDiscovery"
               }
            ]
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
         "type": "structure",
         "members": {
            "SourceRegions": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
               "traits": {
                  "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                  "smithy.api#required": {}
               }
            },
            "OrganizationIntegration": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
               "traits": {
                  "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                  "smithy.api#required": {}
               }
            }
         },
         "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
         "type": "operation",
         "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
         },
         "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
         },
```

```
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
 <code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
 subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
 streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
 </p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
 <li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
 <code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
 <code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
 <code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
```

```
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
   <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
          <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't
```

match a certain subscription's name. \n The valid names for this filter are: \n \n \n \n <code>Subscription</code>\n \n \n \n The valid Operators for this filter are: \n \n \n \n <code>contains</code>\n \n \n \n \n <code>equals</code>\n \n \n \n \n <code>Notequal</code>\n \n \n "
}
},
"MaxResults": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
"traits": {
"smithy.api#documentation": " The maximum items to return in a request. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response. ",
"smithy.api#length": {
"min": 1,
"max": 16384
}
}
}
},
"traits": {
"smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
"type": "structure",
"members": {
"Subscriptions": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
"traits": {
"smithy.api#documentation": " An array that contains subscription objects. "
}
},
"NextToken": {
"target": "smithy.api#String",
"traits": {
"smithy.api#documentation": " The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects. "

```
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",
```

```
    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
            }
          }
        },

        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
        "type": "structure",
        "members": {
          "RegisteredSubscriptionProviders": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
            "traits": {
              "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\tyou specified in the request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
          }
        },
        "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
                "method": "GET",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}

                 }
            }
        },
        "traits": {
            "smithy.api#input": {}
```

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
         },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
```

```
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
              },
              {

        "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
          "type": "list",
          "member": {
            "target": "smithy.api#String"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
          }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
           "traits": {
            "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
            "smithy.api#required": {}
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've
stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat
Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
            "smithy.api#required": {}
          }
        },
        "Tags": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
          "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
          }
        },
        "SubscriptionProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
          }
```

```
            },
            "SubscriptionProviderStatus": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
               }
            }
         },
         "traits": {
            "smithy.api#output": {}
         }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
         "type": "structure",
         "members": {
            "SubscriptionProviderArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
               }
            },
            "SubscriptionProviderSource": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
               "traits": {
                  "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
               }
            },
            "SecretArn": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
               "traits": {
                  "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
               }
            },
            "SubscriptionProviderStatus": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
               "traits": {
                  "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
               }
            },
            "SubscriptionProviderStatusMessage": {
               "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
            }
          },
          "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
          "smithy.api#error": "client"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
```

```
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
```

```
        },
        "value": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {

       "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

       "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
```

```
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
```

```
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
            "resourceArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
                     "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
```

```
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
```

```
            "smithy.api#error": "client"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
          "type": "structure",

       "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tagKeys": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
```

```
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input":
{}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
          "type": "structure",
          "members": {},
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {

  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
              "code": 200,
              "method": "POST",
              "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
          "type": "structure",
```

```
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
              "smithy.api#required":
{}
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>",
              "smithy.api#required": {}
            }
          },
          "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
          }

        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }

      },
          "Status": {
```

```
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
                    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
          }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
             "message": {
                "target": "smithy.api#String"
             }
          },
          "traits": {
             "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
             "smithy.api#error": "client"
          }
       }
    }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
```

ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

```
  * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
```

```
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
```

```
if
err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if
err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
```

```
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
```

```
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
```

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever

InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.104 github.com/aws/aws-sdk-go-v2/feature/s3/manager 1.20.1

1.104.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

```
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
  params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

 // Home Region for the license.
 //
 // This member is required.
 HomeRegion *string

 // License issuer.
 //
 // This member is required.
 Issuer *types.Issuer

 // License name.
 //
 // This member is required.
 LicenseName *string

 // Product name.
 //
 // This member is required.
 ProductName *string

 // Product SKU.
```

```
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
```

```
middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err
= addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
```

```
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return
 err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return
 err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CreateLicense",
 }
}
CheckoutBorrowLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
```

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
  params = &ListReceivedLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListReceivedLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - ProductSKU
 //
 //   - Status
 //
 //
   - Fingerprint
 //
 //   - IssuerName
 //
```

```
//   - Beneficiary
 Filters []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

 // Received license details.
 Licenses []types.GrantedLicense

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
 middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
```

```
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
```

```
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return
 err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if
 err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "ListReceivedLicenses",
 }
}
DeleteLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
```

```
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
 if params == nil {
  params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 //   - Beneficiary
 //
 //   - ProductSKU
 //
 //   - Fingerprint
 //
 //   - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
```

```
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
```

```
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptBeforeRetryLoop(stack,
 options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if
 err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
```

```
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "ListLicenses",
 }
}
DisassociateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
 Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit


```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
```

```
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
          "DomainName": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
          },
          "DomainIpv4List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainIpv6List": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 2
              }
            }
          },
          "DomainCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
```

```
            "traits": {
                "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
 resource that contains information about the credential provider for user administration.</p>"
            }
          },
          "DomainNetworkSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "SELF_MANAGED",
               "value": "SELF_MANAGED",
              "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
              "name": "AWS_MANAGED",
              "value": "AWS_MANAGED",
              "documentation": "AWS_MANAGED type of Active Directory"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target":
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/AssociateUser"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The identity provider for the user.</p>",

           "smithy.api#required": {}
            }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
```

```
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
          "type": "integer"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
          "type": "structure",
          "members": {
            "message": {
              "target": "smithy.api#String"
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
            "smithy.api#error": "server"
          }

 },
        "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },

          {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/CreateLicenseServerEndpoint"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
        "type": "structure",
        "members":
{
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
                    "smithy.api#required": {}
                }
            },
            "LicenseServerSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager
 secret that enables administrators to add or remove users associated with the license server.</p>",
                    "smithy.api#required": {}
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
                }
            }
        },
```

```
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
 provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
 the Active Directory.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
      },
```

```
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/DeleteLicenseServerEndpoint"

         },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
        "type": "structure",
        "members": {
            "LicenseServerEndpointArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                    "smithy.api#required": {}
                }
```

```
            },
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}

  }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
          "type": "structure",
          "members": {
            "LicenseServerEndpoint": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
              "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"

         },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
              "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
```

```
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"

        },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/identity-provider/DeregisterIdentityProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies
 details for the Active Directory identity provider.</p>"
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },

 "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {

       "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {

 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/user/DisassociateUser"
            },
            "smithy.api#idempotent": {}
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
         "type": "structure",
         "members": {
            "Username": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The user name from the Active Directory
 identity provider for the user.</p>"
               }
            },
            "InstanceId": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
               }
            },
            "IdentityProvider": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
               "traits": {
                  "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
```

```
            }
          },
          "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
              "smithy.api#required": {}
            }
          }
        },

      "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
              "smithy.api#length": {
                "min": 1
              },
```

```
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#Filter": {
          "type": "structure",
          "members":
{
            "Attribute": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
              }
            },
            "Operation": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
              }
            },
            "Value": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
 of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Refers to
 an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",

              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
            }

          },
          "FailureMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
              "smithy.api#required": {}
            }
          },
          "Products": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
              "smithy.api#required": {}
            }
          },
          "LastStatusCheckDate": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The date of the last status check.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
            }
          },
          "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
             "traits": {
              "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The user name
 from the identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {

              "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
            }
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
```

```
contains the user information for the product subscription.</p>"
                }
            },
            "AssociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
                }
            },
            "DisassociationDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV4List": {
```

```
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|::|::[0-9a-fA-F]{1,4}|([0-9a-fA-
F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
```

```
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {

      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
         "x-api-key",
        "authorization",
```

```
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager User Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {

    "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
              "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to
 the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
```

```
        },
        "rules": [
          {
            "conditions": [
              {
                "fn": "isSet",
                "argv": [
                  {
                    "ref": "Endpoint"
                  }
                ]

 }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [

{
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      },
                      true
                    ]
                  }
                ],
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                "type": "error"
              },
              {
                "conditions": [],
```

```
            "endpoint": {
              "url": {
                "ref": "Endpoint"
              },

"properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                 "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
```

```
true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {

"fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsFIPS"
                        ]
                      }
                    ]
                  },

                  {
                    "fn": "booleanEquals",
                    "argv": [
                      true,
                      {
                        "fn": "getAttr",
                        "argv": [
                          {
                            "ref": "PartitionResult"
                          },
                          "supportsDualStack"
                        ]
                      }
```

```
                                ]
                                    }
                                ],
                                "rules": [
                                    {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }

            ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [

            {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "fn": "getAttr",
```

```
                        "argv": [

                    {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "endpoint": {

    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS is enabled but this partition does not support FIPS",
        "type": "error"

}
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                "fn": "booleanEquals",
                "argv": [
                    {
                    "ref": "UseDualStack"
                    },
                    true
                ]
                }
            ],
            "rules": [
```

```
                {

        "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        true,
                        {
                          "fn": "getAttr",
                          "argv": [
                            {
                              "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                          ]

    }
                      ]
                    }
                  ],
                  "rules": [
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }

                  ],
                  "type": "tree"
                },
                {
                  "conditions": [],
                  "error": "DualStack is enabled but this partition does not support DualStack",
                  "type": "error"
                }
              ],
              "type": "tree"
            },
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
```

```
                    "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                  }
                ],
                "type": "tree"
              }
            ],
            "type": "tree"
          },
          {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
          }
        ]
      },
      "smithy.rules#endpointTests": {
        "testCases": [
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
              }
            },
            "params": {
              "Region": "us-east-1",
              "UseFIPS": true,
              "UseDualStack": true
            }
          },
          {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
              }
            },
            "params": {
              "Region": "us-east-1",

        "UseFIPS": true,
              "UseDualStack": false
            }
```

```
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

  "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
 true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
```

```
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
          "expect": {

"endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
          },
          "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
```

```
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
```

```
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
          },
          "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
          "expect": {

      "endpoint": {
              "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
          },
          "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
```

```
        },
        {
          "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,

    "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
          "expect": {
            "endpoint": {
              "url": "https://example.com"
            }
          },
          "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips enabled and dualstack disabled",
          "expect": {

  "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
          }
        },
        {
          "documentation": "For custom endpoint with fips disabled and dualstack enabled",
          "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
          },
```

```
                "params": {
                  "Region": "us-east-1",
                  "UseFIPS": false,
                  "UseDualStack": true,
                  "Endpoint": "https://example.com"
                }
              },
              {
                "documentation": "Missing region",
                "expect": {
                  "error": "Invalid Configuration: Missing Region"
                }
              }
            ],
            "version": "1.0"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
        "type": "structure",
        "members": {
          "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
              "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
          },
          "HealthStatus": {
            "target":
 "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
              "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
          },
          "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
          },
          "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
          "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's associated with the RDS license server endpoint.</p>"
            }
          },
          "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
              "smithy.api#documentation": "<p>The type of license server.</p>"
            }
          },
          "ServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
            }
          },
          "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is one.</p>"
            }
          },
          "LicenseServerEndpointId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
            }
          },
          "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
```

```
 resource for the RDS license server.</p>"
                }
            },
            "LicenseServerEndpointProvisioningStatus": {
                "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
                }
            },
            "LicenseServers": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
                }
            },
            "CreationTime": {
                "target": "smithy.api#Timestamp",

    "traits": {
                    "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
```

```
            "name": "PROVISIONING",
              "value": "PROVISIONING",
              "documentation": ""
            },
            {
              "name": "PROVISIONING_FAILED",
              "value": "PROVISIONING_FAILED",
              "documentation": ""
            },
            {
              "name": "PROVISIONED",
              "value": "PROVISIONED",
              "documentation": ""
            },
            {
              "name": "DELETING",
              "value": "DELETING",
              "documentation": ""
            },
            {
              "name": "DELETION_FAILED",
              "value": "DELETION_FAILED",
              "documentation": ""
            },
            {
              "name": "DELETED",

"value": "DELETED",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "HEALTHY",
              "value": "HEALTHY",
              "documentation": ""
            },
            {
              "name": "UNHEALTHY",
              "value": "UNHEALTHY",
              "documentation": ""
            },
            {
```

```
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
          "type": "list",
           "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
          "type": "structure",
          "members": {
            "ServerType": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
              "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
              }
            },
            "ServerSettings": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
              }

  }
        },
        "traits": {
          "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
```

```
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
             {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
               "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
         ],
         "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
               "code": 200,
               "method": "POST",
               "uri": "/identity-provider/ListIdentityProviders"
            },
            "smithy.api#paginated": {
               "inputToken": "NextToken",
                "outputToken": "NextToken",
               "pageSize": "MaxResults",
               "items": "IdentityProviderSummaries"
            }
         }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
         "type": "structure",
         "members": {
            "MaxResults": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
               "traits": {
                  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
               }
            },
            "Filters": {
               "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
```

```
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources that contain details about the Active Directory identity providers that meet the request criteria.</p>",
              "smithy.api#required": {}
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
        },
        "output": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "InstanceSummaries"
        }
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
      "type": "structure",
      "members": {
        "MaxResults": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
          "traits": {
            "smithy.api#documentation": "<p>The maximum number of results
```

```
 to return from a single request.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
                }
            },
            "Filters": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
                "traits": {
                    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
        "type": "structure",

     "members": {
            "InstanceSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
     "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
```

```
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },

      {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/license-server/ListLicenseServerEndpoints"
            },
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "LicenseServerEndpoints"
            }
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
        "members": {
            "MaxResults": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>",
              "smithy.api#range": {
                "min": 1,
                "max": 100
              }
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>IdentityProviderArn</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {

          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
          "LicenseServerEndpoints": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
              "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits":
 {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
```

```
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },

            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/user/ListProductSubscriptions"
            },
            "smithy.api#paginated": {
```

```
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "ProductUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
 </p>"
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline
 results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
          }
        },
```

```
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
 operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
 empty, there are additional elements that the service hasn't included in this request. Use this token with the next
 request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"

 },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
```

```
            "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{ResourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "ResourceArn": {
              "target":
 "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}

 }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
             "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
          },
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceUserSummaries"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
        "type": "structure",
        "members": {
          "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
              "smithy.api#required": {}
```

```
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
              "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
          },
          "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }

 },
     "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
          "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
```

```
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#output":
 {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
              "smithy.api#required": {}
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
              "smithy.api#required": {}
            }
          },
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {

  "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
              "smithy.api#required": {}
            }
          },
          "Status": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The status of a product for this user.</p>",
              "smithy.api#required": {}
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
```

```
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
              }
            },
            "StatusMessage": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The status message for a
 product for this user.</p>"
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
 information for the product subscription.</p>"
              }
            },
            "SubscriptionStartDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The start date of a subscription.</p>"
              }
            },
            "SubscriptionEndDate": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The end date of a subscription.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>A summary
 of the user-based subscription products for a specific user.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
          "type": "structure",
          "members": {
            "RdsSalCredentialsProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
```

reference to the credentials provider that's used for RDS license server user administration. ",
"smithy.api#required": {}
}
}
},
"traits": {

"smithy.api#documentation": " Server settings that are specific to a Remote Desktop Services (RDS) license server. "
}
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
"type": "operation",
"input": {
"target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
},
"errors": [
{
"target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
"target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
"smithy.api#documentation": " Registers an identity provider for user-based subscriptions. ",
"smithy.api#http": {
"code": 200,
"method": "POST",
"uri": "/identity-provider/RegisterIdentityProvider"

```
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                    "smithy.api#required": {}
                }

            },
            "Settings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
```

```
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"

          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
          "smithy.api#error": "client",
          "smithy.api#httpError": 404
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
          "SecretId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
              "smithy.api#length": {
                "min": 1
              }
            }
          }
        },
```

```
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
        "type": "union",

     "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
```

```
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
      "type": "structure",
      "members": {
        "Subnets": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
          "traits": {
            "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
            "smithy.api#length": {
              "min": 1
            },
            "smithy.api#required": {}
          }

    },
        "SecurityGroupId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
          "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC. "
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":
{
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": " Starts a product subscription for a user with the specified identity provider. <note> Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing. For more information, see Viewing your monthly charges in the Amazon Web Services Billing User Guide. </note>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",

```
 "uri": "/user/StartProductSubscription"
            }
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
          "type": "structure",
          "members": {
            "Username": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
              }
            },
            "IdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
              }
            },
            "Product": {
              "target": "smithy.api#String",
               "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
              }
            },
            "Domain": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
              }
            },
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
```

```
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
```

```
        ],
        "traits": {
          "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
          }
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
        "type": "structure",
        "members": {
          "Username": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
          },
          "IdentityProvider":
{
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
              "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "ProductUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
             }
          },
          "Domain": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
            }
          }
```

```
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
          "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
              "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
              "smithy.api#required": {}
            }

       }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
```

```
      "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
          },
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{ResourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "ResourceArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
              "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "Tags": {
```

```
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
              "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#Tags": {
        "type": "map",
        "key": {
          "target": "smithy.api#String"
        },
        "value": {
          "target": "smithy.api#String"
        },
        "traits": {
          "smithy.api#length": {
            "min": 0,
            "max": 50
          },
          "smithy.api#sensitive": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
          "smithy.api#error": "client"
        }
```

```
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {

         "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

        }
        },
        "TagKeys": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
          "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
          }
        }
```

```
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
            "smithy.api#http": {

  "code": 200,
                "method": "POST",
                "uri": "/identity-provider/UpdateIdentityProviderSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
```

```
        "members": {
          "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
          },
          "Product": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
          },
          "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
          },
          "UpdateSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
            "traits": {
              "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li> <p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
              "smithy.api#required": {}
            }

          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        },
        "traits": {
```

```
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one
 or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                    "smithy.api#required": {}
                }
            },
            "RemoveSubnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                    "smithy.api#required": {}
                }
            },
            "SecurityGroupId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
                "traits": {
                    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
                }
            }
        },

        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
```

```
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
  params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DisassociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}
```

```
type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
 err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
```

```
 return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if
 err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
```

```
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if
```

```
 err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "DisassociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
  params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // License version.
 Version *string

 noSmithyDocumentSerde
}

type
 GetLicenseOutput struct {

 // License details.
 License *types.License

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err
```

```
!= nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
 }
```

```
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:
       region,
  ServiceID:     ServiceID,
  OperationName: "GetLicense",
 }
}
name: License Scan

on: [pull_request]


permissions:
 contents: read

jobs:
 licensescan:
   name: License Scan
   runs-on: ubuntu-latest
   strategy:
     matrix:
       python-version: [3.9]

   steps:
     - name: Checkout target
       uses: actions/checkout@v2
       with:
         path: sdkbase
         ref: ${{ github.base_ref }}
```

```
    - name: Checkout this ref
      uses: actions/checkout@v2
      with:
        path: new-ref
        fetch-depth: 0
    - name: Get Diff
      run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
    - name: Get Target Files
      run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
    - name: Checkout scancode
      uses: actions/checkout@v2
      with:
        repository: nexB/scancode-toolkit
        path: scancode-toolkit
        fetch-depth: 1
    - name: Set up Python
 ${{ matrix.python-version }}
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    # ScanCode
    - name: Self-configure scancode
      working-directory: ./scancode-toolkit
      run: ./scancode --help
    - name: Run Scan code on target
      run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
    - name: Run Scan code on pr ref
      run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
    # compare
    - name: License test
      run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
  params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

 // Product SKU.
 //
 // This member is required.
```

```
 ProductSKU *string

 // License beneficiary.
 Beneficiary *string

 // Node ID.
 NodeId *string

 noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

 // Checkout type.
 CheckoutType types.CheckoutType

 // Allowed license entitlements.
 EntitlementsAllowed []types.EntitlementData

 // Date and time at which the license checkout expires.
 Expiration *string

 // Date and time at which the license checkout is issued.
 IssuedAt *string

 // Amazon Resource Name (ARN) of the checkout license.
 LicenseArn
 *string

 // License consumption token.
 LicenseConsumptionToken *string

 // Node ID.
 NodeId *string

 // Signed token.
 SignedToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
```

```
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
 return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
```

```
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack,
 c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
  return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
  return err
```

```
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "CheckoutLicense",
 }
}
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
  params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod
```

```
// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
```

```
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
 if err != nil {
  return
 err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
```

```
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
 return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
 err != nil {
 return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
 middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options);
 err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
```

```
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}
```

```
func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
  OperationName: "CheckoutBorrowLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
  params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
```

```
//
// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
 return err
}
```

```
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addlegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
```

```
c); err != nil {
 return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
 }
 if err = addCredentialSource(stack, options); err != nil {
 return err
 }
 if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
 }
 if err = addRecursionDetection(stack); err != nil {
 return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
 return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
 return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addInterceptAttempt(stack, options); err != nil {
 return err
 }
 if err = addInterceptExecution(stack,
 options); err != nil {
 return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
```

```
 return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:        region,
  ServiceID:     ServiceID,
  OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
  params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //
 // This member is required.
 SourceVersion
 *string

 noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
```

```
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err !=
 nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
```

```
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
 != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
 if err = addResponseErrorMiddleware(stack); err != nil {
  return err
 }
 if err = addRequestResponseLogging(stack, options); err != nil {
  return err
 }
 if err = addDisableHTTPSMiddleware(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
  return err
 }
```

```
 if err = addInterceptAttempt(stack, options); err !=
 nil {
  return err
 }
 if err = addInterceptExecution(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSerialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterSigning(stack, options); err != nil {
  return err
 }
 if err = addInterceptTransmit(stack, options); err != nil {
  return err
 }
 if err = addInterceptBeforeDeserialization(stack, options); err != nil {
  return err
 }
 if err = addInterceptAfterDeserialization(stack, options); err != nil {
  return err
 }
 if err = addSpanInitializeStart(stack); err != nil {
  return err
 }
 if err = addSpanInitializeEnd(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestStart(stack); err != nil {
  return err
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
 string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:    ServiceID,
```

```
 OperationName: "DeleteLicense",
 }
}
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },

    {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own
License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
```

```
        "members": {
          "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
             "traits": {
               "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
               "smithy.api#required": {}
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {},
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
        "type": "structure",
        "members": {
          "Name": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of name to filter by.</p>"
             }
          },
          "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
              "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
          },
          "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
              "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n      Filters can be used to match a set of resources by specific criteria.</p>"
        }
      },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
 "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
         },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
```

```
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

          "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderStatus": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
                 }
            },
            "SubscriptionProviderStatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
                }
            },
```

```
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"

        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
```

```
        "type": "structure",
        "members": {
          "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
              "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
          },
          "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
              "smithy.api#documentation": "<p>Lists
 the settings defined for Linux subscriptions discovery. The settings include if\n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
            }
          },
          "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
          },
          "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
              "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
          },
          "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
             "traits": {
              "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
        "type": "structure",
        "members": {
          "AmiId": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
          },
```

```
            "InstanceID": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
               }
            },
            "InstanceType": {
               "target": "smithy.api#String",
               "traits":
{
                  "smithy.api#documentation": "<p>The instance type of the resource.</p>"
               }
            },
            "AccountID": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
               }
            },
            "Status": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The status of the instance.</p>"
               }
            },
            "Region": {
               "target": "smithy.api#String",
               "traits": {
                  "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
               }
            },
            "UsageOperation": {
               "target": "smithy.api#String",

   "traits": {
                  "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n        <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i>.</p>"
               }
            },
            "ProductCode": {
               "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
               "traits": {
                  "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
               }
```

```
        },
        "LastUpdatedTime": {
          "target": "smithy.api#String",
           "traits": {
             "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
          }
        },
        "SubscriptionName": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
          }
        },
        "OsVersion": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
          }
        },
        "SubscriptionProviderCreateTime": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The timestamp when you registered
 the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
          }
        },
        "SubscriptionProviderUpdateTime": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
          }
        },
        "DualSubscription": {
          "target": "smithy.api#String",
          "traits": {
             "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
          }
        },
        "RegisteredWithSubscriptionProvider": {
          "target": "smithy.api#String",
          "traits": {

    "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation":
 "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
        "type": "service",
        "version": "2018-05-10",
        "operations": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
```

```
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
      ],
      "traits": {
        "aws.api#service": {
          "sdkId": "License Manager Linux Subscriptions",
           "arnNamespace": "license-manager-linux-subscriptions",
          "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
        },
        "aws.auth#sigv4": {
          "name": "license-manager-linux-subscriptions"
        },
        "aws.protocols#restJson1": {},
        "smithy.api#cors": {
          "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id"
          ],
          "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
          ]
        },
        "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
        "smithy.api#title": "AWS License Manager Linux Subscriptions",
        "smithy.rules#endpointRuleSet": {
          "version": "1.0",
          "parameters": {
```

```
            "Region": {
              "builtIn": "AWS::Region",
              "required": false,
              "documentation": "The AWS region used to dispatch the request.",
              "type": "string"
            },
            "UseDualStack": {
              "builtIn": "AWS::UseDualStack",
              "required": true,
              "default": false,
              "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
               "type": "boolean"
            },
            "UseFIPS": {
              "builtIn": "AWS::UseFIPS",
              "required": true,
              "default": false,
              "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
              "type": "boolean"
            },
            "Endpoint": {
              "builtIn": "SDK::Endpoint",
              "required": false,
              "documentation": "Override the endpoint used to send this request",
              "type": "string"
            }
          },
          "rules": [
            {
              "conditions": [
                {
                  "fn": "isSet",
                  "argv": [
                    {
                      "ref": "Endpoint"
                    }
                  ]
                }
              ],
              "rules": [
                {
                  "conditions": [
                    {
                      "fn": "booleanEquals",
                      "argv": [
                        {
```

```
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }

    ],
                  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
                  "type": "error"
                },
                {
                  "conditions": [],
                  "rules": [
                    {
                      "conditions": [
                        {
                          "fn": "booleanEquals",
                          "argv": [
                            {
                              "ref": "UseDualStack"
                            },
                            true
                          ]

}
                      ],
                      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
                      "type": "error"
                    },
                    {
                      "conditions": [],
                      "endpoint": {
                        "url": {
                          "ref": "Endpoint"
                        },
                        "properties": {},
                        "headers": {}
                      },
                      "type": "endpoint"
                    }
                  ],
                  "type": "tree"

              }
              ],
              "type": "tree"
            },
            {
```

```
"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {

"fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ],
            "assign": "PartitionResult"
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {

                "ref": "UseFIPS"
                  },
                  true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
```

```
                }

    ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {
                                                "ref": "PartitionResult"

                    },
                                        "supportsFIPS"
                                    ]
                                }
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                true,
                                {
                                    "fn": "getAttr",
                                    "argv": [

                                     {
                                            "ref": "PartitionResult"
                                        },
                                        "supportsDualStack"
                                    ]
                                }
                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],

        "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
```

```
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                                "properties": {},
                                                "headers": {}
                                            },
                                            "type": "endpoint"
                                        }
                                    ],

                            "type": "tree"
                                    }
                                ],
                                "type": "tree"
                            },
                            {
                                "conditions": [],
                                "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
                                "type": "error"
                            }
                        ],
                        "type": "tree"
                    },
                    {

            "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseFIPS"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {

                        "fn": "booleanEquals",
                                        "argv": [
                                            {
                                                "fn": "getAttr",
                                                "argv": [
                                                    {
                                                        "ref": "PartitionResult"
                                                    },
```

```
                                        "supportsFIPS"
                                    ]
                                },
                                true

                            ]
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

    "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"
                        }
                    ],
                    "type": "tree"
                },
                {
                    "conditions": [],
                    "error": "FIPS is enabled
but this partition does not support FIPS",
                    "type": "error"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
```

```
                ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    true,
                                    {
                                        "fn": "getAttr",
                                        "argv": [
                                            {

                                            "ref": "PartitionResult"
                                            },
                                            "supportsDualStack"
                                        ]
                                    }
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [],
                                "rules": [

        {
                                        "conditions": [],
                                        "endpoint": {
                                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                            "properties": {},
                                            "headers": {}
                                        },
                                        "type": "endpoint"
                                    }
                                ],
                                "type": "tree"

                    }
                        ],
                        "type": "tree"
                    },
                    {
                        "conditions": [],
                        "error": "DualStack is enabled but this partition does not support DualStack",
```

```
                                    "type": "error"
                                }
                            ],
                            "type": "tree"
                        },
                        {
                            "conditions": [],
                            "rules": [
                                {
                                    "conditions": [],
                                    "endpoint": {
                                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                                        "properties": {},
                                        "headers": {}
                                    },
                                    "type": "endpoint"
                                }
                            ],
                            "type": "tree"
                        }
                    ],
                    "type": "tree"
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "Invalid Configuration: Missing Region",
            "type": "error"
        }
    ],
    "type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

}
            },
            "params": {
```

```
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
          }
        },
        {

      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
          },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
          }
        },
        {
          "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
          "expect": {
            "endpoint": {
              "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }

    },
          "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
          }
        },
        {
          "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
```

```
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            {

 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                    }
                 },
                "params": {
                    "Region": "cn-north-1",
                    "UseFIPS": false,
                    "UseDualStack": true
                }
            },
            {
                "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
                    }
                },
                "params": {
                    "Region": "cn-north-1",
```

```
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }

     },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
```

```
"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
                      }
                    },
                    "params": {
                      "Region": "us-gov-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
                      }

                  },
                    "params": {
                      "Region": "us-iso-east-1",
                      "UseFIPS": true,
                      "UseDualStack": false
                    }
                  },
                  {
                    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
                      }
                    },
                    "params": {
                      "Region": "us-iso-east-1",
                      "UseFIPS": false,
                      "UseDualStack": false
                    }
                  },
                  {

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
                    "expect": {
                      "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
                      }
                    },
                    "params": {
```

```
                "Region": "us-isob-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"

    }
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },

        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
```

```
            {
              "documentation": "For custom endpoint with fips enabled and dualstack disabled",
              "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
              },
              "params": {

      "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "For custom endpoint with fips disabled and dualstack enabled",
              "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
              },
              "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
              }
            },
            {
              "documentation": "Missing region",

            "expect": {
                "error": "Invalid Configuration: Missing Region"
              }
            }
          ],
          "version": "1.0"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "Enabled",
            "value": "Enabled",
            "documentation": "Enabled LinuxSubscriptionsDiscovery"
          },
          {
            "name": "Disabled",
```

```
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
              }
            ]
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
          "type": "structure",
          "members": {
            "SourceRegions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
              "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
              }
            },
            "OrganizationIntegration": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
              "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
 subscriptions.</p>"
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
          },
          "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Lists
 the running Amazon EC2 instances that were discovered with commercial Linux\n      subscriptions.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptionInstances"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Instances"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
              "smithy.api#documentation": "<p>An array of structures
 that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
  <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n          <li>\n            <p>\n              <code>AccountID</code>\n
</p>\n          </li>\n          <li>\n            <p>\n              <code>AmiID</code>\n            </p>\n          </li>\n
        <li>\n            <p>\n              <code>DualSubscription</code>\n            </p>\n          </li>\n
<li>\n            <p>\n              <code>InstanceID</code>\n            </p>\n          </li>\n          <li>\n
  <p>\n              <code>InstanceType</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
    <code>ProductCode</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Region</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>Status</code>\n            </p>\n          </li>\n          <li>\n            <p>\n
<code>UsageOperation</code>\n            </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n  \t\tdefine the behavior of the filter:</p>\n        <ul>\n
    <li>\n            <p>\n              <code>contains</code>\n            </p>\n          </li>\n          <li>\n
<p>\n              <code>equals</code>\n            </p>\n          </li>\n          <li>\n
            <p>\n              <code>Notequal</code>\n            </p>\n          </li>\n        </ul>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
```

```
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
              "smithy.api#length": {
                "min": 1,
                "max": 16384
              }
            }
          }
        },
        "traits": {
          "smithy.api#documentation":
 "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
          "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
              "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
          },
          "NextToken": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
 objects.</p>"
            }
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
```

```
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n      organization, the returned results will include data aggregated across your accounts in\n      Organizations.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
        "type": "structure",
        "members": {
          "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n      more sets of key-value pairs that you specify. For example, you can filter by the name of\n      <code>Subscription</code> with an optional operator to see subscriptions that match,\n      partially match, or don't match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>Subscription</code>\n      </p>\n      </li>\n      </ul>\n      <p>The valid Operators for this filter are:</p>\n      <ul>\n      <li>\n      <p>\n      <code>contains</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>equals</code>\n      </p>\n      </li>\n      <li>\n      <p>\n      <code>Notequal</code>\n      </p>\n      </li>\n      </ul>"
            }
          },
```

```
            "MaxResults": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
              "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                  "min": 1,
                  "max": 16384
                }
              }
            }
          },
          "traits": {
            "smithy.api#documentation": "NextToken
 length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
          "type": "structure",
          "members": {
            "Subscriptions": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
              "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"

              }
            }
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
          "type": "operation",
          "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
```

```
        },
        "output": {
          "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation":
 "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
          "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
          },
          "smithy.api#idempotent": {},
          "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

    "traits": {
                "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
            }
          },
          "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
              "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
```

```
                "smithy.api#range": {
                  "min": 1,
                  "max": 100
                }
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
              }
            }
          },

          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
          "type": "structure",
          "members": {
            "RegisteredSubscriptionProviders": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
              "traits": {
                "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria
\n\t\t\tyou specified in the request.</p>"
              }
            },
            "NextToken": {
              "target": "smithy.api#String",
              "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
 the next request to retrieve \n\tadditional objects.</p>"
              }
            }
          },
          "traits": {
            "smithy.api#output": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
          "type": "operation",
          "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
          },
          "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
             }
          ],
          "traits": {
            "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
            "smithy.api#http": {
              "method": "GET",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#readonly": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
          "type": "structure",
          "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

              }
            }
          },
          "traits": {
            "smithy.api#input": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
          "type": "structure",
          "members": {
            "tags": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
```

```
            }
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "EQUAL",
              "value": "Equal",
              "documentation": "Equal operator"
             },
            {
              "name": "NOT_EQUAL",
              "value": "NotEqual",
              "documentation": "Not equal operator"
            },
            {
              "name": "CONTAINS",
              "value": "Contains",
              "documentation": "Contains operator"
            }
          ],
          "smithy.api#length": {
            "min": 1,
            "max": 20
          }
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "Enabled",
              "value": "Enabled",
              "documentation": "Enabled OrganizationIntegration"
            },
            {

      "name": "Disabled",
              "value": "Disabled",
              "documentation": "Disabled OrganizationIntegration"
            }
```

```
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },

           {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring
Your Own License (BYOL) subscription.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/RegisterSubscriptionProvider"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                 "traits": {
                    "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
                    "smithy.api#required": {}
```

```
            }
          },
          "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the secret where you've stored your subscription provider's access token. For \n\t\t\tRHEL subscriptions managed through the Red Hat Subscription Manager (RHSM), the secret contains \n\t\t\tyour Red Hat Offline token.</p>",
              "smithy.api#required": {}
            }
          },
          "Tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {

           "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription provider \n\t\t\tresource.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
          "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
              "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
          },
          "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
 the Linux subscription provider resource that you registered.</p>"
            }
          },
          "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
              "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\tthat you requested.</p>"
            }
          }
        },
```

```
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
 Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
```

```
            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",
            "documentation": "InProgress status"
          },
          {
            "name": "Completed",
```

```
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
     },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
```

```
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
          },
          "Type": {
            "target": "smithy.api#String",
            "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
            }
          },
          "InstanceCount": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
            "traits": {
              "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

       "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
```

```
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "ACTIVE",
                    "value": "ACTIVE",
                    "documentation": "ACTIVE status"
                },
                {
                    "name": "INVALID",
                    "value": "INVALID",
                    "documentation": "INVALID status"
                },
                {
                    "name": "PENDING",
                    "value": "PENDING",
                    "documentation": "PENDING status"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
```

```
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"

          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
          }
        ],
        "traits": {
          "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
          "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
          },
          "smithy.api#idempotent": {}
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
        "type": "structure",
        "members": {
          "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
              "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\tmetadata tags.</p>",
               "smithy.api#httpLabel": {},
              "smithy.api#required": {}
            }
          },
          "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
              "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
              "smithy.api#required": {}
            }
```

```
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
        "type": "map",
        "key":
{
            "target": "smithy.api#String"
        },
        "value": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
         },
        "output": {
```

```
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
          },
          "errors": [
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
          ],
          "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
            "smithy.api#http": {
              "method": "DELETE",
              "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
          }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
          "type": "structure",

      "members": {
            "resourceArn": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
              "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
              }
            },
            "tagKeys": {
              "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
              "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
              }
            }
          },
          "traits": {
            "smithy.api#input":
 {}
          }
        },
```

```
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {

 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
            "smithy.api#required":
{}
          }
        },
```

```
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>",
                    "smithy.api#required": {}
                }
            },
            "AllowUpdate": {
                "target": "smithy.api#Boolean",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n      allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
                }
            }

        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n      integration has been enabled, and which Regions data will be aggregated from.</p>"
                }

    },
            "Status": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
                "traits": {
                    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
                }
            },
            "StatusMessage": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
                "traits": {
```

```
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
                }
            },
            "HomeRegions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
                }
             }
          }
       },
       "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
          "type": "structure",
          "members": {
             "message": {
                "target": "smithy.api#String"
             }
          },
          "traits": {
             "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
             "smithy.api#error": "client"
          }
       }
    }
}
CheckInLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
```

AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

```
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
  params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
 if err != nil {
  return nil, err
 }

 out := result.(*CheckInLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string
```

```
 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
 CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
  return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
  return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
  return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addlegacyEndpointContextSetter(stack, options); err != nil {
  return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
  return err
 }
 if err = addClientRequestID(stack); err != nil {
  return err
 }
 if
 err = addComputeContentLength(stack); err != nil {
  return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
  return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
  return err
```

```
 }
 if err = addRetry(stack, options); err != nil {
  return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
  return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
  return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
  return err
 }
 if err = addClientUserAgent(stack, options); err != nil {
  return err
 }
 if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
  return err
 }
 if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
  return err
 }
 if err = addTimeOffsetBuild(stack, c); err != nil {
  return err
 }
 if err = addUserAgentRetryMode(stack, options); err != nil {
  return err
 }
 if
 err = addCredentialSource(stack, options); err != nil {
  return err
 }
 if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
  return err
 }
 if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
  return err
 }
 if err = addRecursionDetection(stack); err != nil {
  return err
 }
 if err = addRequestIDRetrieverMiddleware(stack); err != nil {
  return err
 }
```

```
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
 return err
}
if err = addInterceptAttempt(stack, options); err != nil {
 return err
}
if err = addInterceptExecution(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
 return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
 return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
 return err
}
if err = addInterceptTransmit(stack, options); err != nil {
 return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
 return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
```

```
 }
 if err = addSpanBuildRequestEnd(stack); err != nil {
  return err
 }
 return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
  Region:       region,
  ServiceID:
     ServiceID,
  OperationName: "CheckInLicense",
 }
}
CreateLicense
 Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
 Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
 Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
 Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
AssociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step

spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter

SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries.To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a

partnership relationship between Cisco and any other company. (1110R)

©2025 Cisco Systems, Inc. All rights reserved.